



---

# dashBIRD

dashBIRD Referenz

---

Dokumentversion vom 10.01.2025

# Inhalt

<b>1</b>	<b>C3 Charts [Options]</b> .....	<b>319</b>
<b>2</b>	<b>Dashlet Engine [Options]</b> .....	<b>320</b>
2.1	Dashlet Engine Buttons .....	320
2.1.1	button.onclick .....	320
2.2	Dashlet Engine Data .....	321
2.2.1	data.script.dimZ1Filter .....	321
2.2.2	data.script.template .....	321
2.2.3	chart.predefined.xAxis .....	321
2.2.4	data.script.searchOperator1 .....	322
	Erlaubte Werte .....	322
2.2.5	data.script.searchOperator2 .....	323
	Erlaubte Werte .....	323
2.2.6	data.script.searchOperator3 .....	323
	Erlaubte Werte .....	324
2.2.7	data.script.searchOperator4 .....	324
	Erlaubte Werte .....	325
2.2.8	data.script.searchOperator5 .....	325
	Erlaubte Werte .....	325
2.2.9	data.script.mode .....	326
	Erlaubte Werte .....	326
2.2.10	data.script.dimZ0Filter .....	326
2.2.11	data.script.dimZ2Filter .....	327
2.2.12	data.script.dimZ3Filter .....	327
2.2.13	data.script.dimZ4Filter .....	328
2.2.14	data.script.dimZ5Filter .....	328
2.2.15	chart.series.data.column .....	329
2.2.16	chart.series.data.function .....	329
2.2.17	data.script.columnSearch .....	330
2.2.18	data.script.columnSearchColumns .....	330
2.2.19	data.script.flexselect .....	331
2.2.20	peity.chart.pie.fill .....	331
2.2.21	peity.chart.pie.radius .....	332
2.2.22	peity.chart.line.delimiter .....	332

2.2.23	peity.chart.line.fill.....	332
2.2.24	peity.chart.line.height .....	333
2.2.25	peity.chart.line.min.....	333
2.2.26	peity.chart.line.stroke .....	334
2.2.27	peity.chart.line.strokeWidth.....	334
2.2.28	peity.chart.line.width .....	335
2.2.29	peity.chart.bar.delimiter .....	335
2.2.30	peity.chart.bar.fill .....	335
2.2.31	peity.chart.bar.height.....	336
2.2.32	peity.chart.bar.min .....	336
2.2.33	peity.chart.bar.width .....	337
2.2.34	peity.chart.bar.padding.....	337
2.3	<b>Dashlet Engine General.....</b>	<b>338</b>
2.3.1	html.after.content.....	338
2.3.2	html.before.content.....	338
2.3.3	box.info.icon.....	338
2.3.4	box.info.tooltip.show.....	339
	Erlaubte Werte .....	339
2.3.5	box.info.tooltip.title.....	339
2.3.6	box.info.tooltip.delay .....	340
2.3.7	box.info.tooltip.template .....	340
2.3.8	box.info.popover.show.....	340
	Erlaubte Werte .....	341
2.3.9	box.info.popover.placement.....	341
2.3.10	box.info.popover.trigger.....	341
2.3.11	box.info.popover.offset .....	342
2.3.12	box.info.popover.delay.....	342
2.3.13	box.info.popover.html.....	342
2.3.14	box.info.popover.title .....	343
2.3.15	box.info.popover.content.....	343
2.3.16	box.info.dialog.html .....	343
2.3.17	box.info.dialog.content .....	344
2.3.18	box.download.native.link.....	344
2.3.19	bar.n[].component.n[].preselect.....	345
2.3.20	chart.predefined.legend.....	345
2.3.21	chart.custom.function .....	346

2.3.22	childid .....	346
2.3.23	javascript .....	347
2.3.24	chart.event.click.....	347
2.3.25	table.lazy.load .....	348
2.3.26	table.lazy.load.time .....	348
2.3.27	chart.stacked.....	348
2.3.28	chart.connect .....	349
2.3.29	chart.connect.time.init .....	349
2.3.30	chart.connect.time.retry .....	350
2.3.31	chart.series.type.....	350
2.3.32	chart.map.type.....	351
2.3.33	chart.map.contextMenu.show .....	351
2.3.34	chart.event.contextmenu .....	352
2.3.35	dashlet.external.json .....	352
2.3.36	box.closelink.show .....	353
	Erlaubte Werte .....	353
2.3.37	box.collapselink.show .....	353
	Erlaubte Werte .....	353
2.3.38	child.box .....	353
2.3.39	datatable.id .....	354
2.3.40	datatable.column.seriesfilter .....	355
2.3.41	magic.add.buttonbar.....	355
2.3.42	magic.buttonbar.event.onclick.....	356
2.3.43	chart.inline.mode.....	356
2.3.44	bar.n[].component.n[].options.var .....	356
2.3.45	bar.n[].component.onSelectAll.....	357
2.3.46	bar.n[].component.onChange.....	357
2.3.47	bar.n[].component.onDeselectAll.....	358
2.3.48	bar.n[].block.....	358
	Erlaubte Werte .....	359
2.3.49	bar.n[].component.n[].maxHeight.....	359
2.3.50	bar.n[].component.n[].enableFiltering.....	359
2.3.51	bar.n[].component.n[].enableCaseInsensitiveFiltering.....	360
2.3.52	bar.n[].component.n[].enableFullValueFiltering .....	360
2.3.53	bar.n[].component.n[].filterBehavior .....	361
2.3.54	bar.n[].component.n[].filterPlaceholder .....	361



2.3.55	bar.n[].component.n[].buttonText .....	362
2.3.56	bar.n[].component.n[].buttonTitle .....	363
2.3.57	bar.n[].component.n[].nonSelectedText.....	363
2.3.58	bar.n[].component.n[].nSelectedText .....	364
2.3.59	bar.n[].component.n[].allSelectedText .....	364
2.3.60	bar.n[].component.n[].optionClass .....	364
2.3.61	bar.n[].component.n[].selectedClass .....	365
2.3.62	bar.n[].component.n[].includeSelectAllOption .....	365
2.3.63	bar.n[].component.n[].enableClickableOptGroups.....	366
2.3.64	bar.n[].component.n[].enableCollapsibleOptGroups.....	366
2.3.65	bar.n[].component.n[].disableIfEmpty.....	366
2.3.66	bar.n[].component.n[].disabledText .....	367
2.3.67	bar.n[].component.n[].dropRight .....	367
2.3.68	bar.n[].component.n[].dropUp .....	367
2.3.69	bar.n[].component.n[].checkboxName .....	368
2.3.70	bar.n[].component.n[].onInitialized .....	368
2.3.71	bar.n[].component.n[].onDropdownShow.....	368
2.3.72	bar.n[].component.n[].onDropdownHide .....	369
2.3.73	bar.n[].component.n[].onDropdownShown .....	369
2.3.74	bar.n[].component.n[].onDropdownHidden.....	369
2.3.75	bar.n[].component.n[].buttonClass .....	370
2.3.76	bar.n[].component.n[].inheritClass .....	370
2.3.77	bar.n[].component.n[].buttonContainer .....	370
2.3.78	bar.n[].component.n[].buttonWidth.....	371
2.3.79	bar.n[].component.n[].numberDisplayed .....	371
2.3.80	bar.n[].component.n[].delimiterText.....	372
2.3.81	bar.n[].component.n[].optionLabel.....	372
2.3.82	bar.n[].component.n[].selectAllJustVisible.....	373
2.3.83	bar.n[].component.n[].selectAllText.....	373
2.3.84	bar.n[].component.n[].selectAllValue.....	373
2.3.85	bar.n[].component.n[].selectAllName .....	374
2.3.86	bar.n[].component.n[].selectAllNumber .....	374
2.3.87	bar.n[].style .....	375
2.3.88	chart.height .....	375
2.3.89	table.customScript .....	375
2.3.90	box.info.dialog.show .....	376

	Erlaubte Werte .....	376
2.3.91	box.info.dialog.title.....	376
2.3.92	box.info.dialog.type .....	377
	Erlaubte Werte .....	377
2.3.93	box.info.dialog.footer.show .....	377
	Erlaubte Werte .....	378
2.3.94	box.info.dialog.footer.button.close.title .....	378
2.3.95	box.alert.dialog.show .....	378
	Erlaubte Werte .....	379
2.3.96	box.alert.dialog.title .....	379
2.3.97	box.alert.dialog.type.....	379
	Erlaubte Werte .....	380
2.3.98	box.alert.dialog.content .....	380
2.3.99	box.alert.dialog.footer.show .....	380
	Erlaubte Werte .....	380
2.3.100	box.alert.dialog.footer.button.close.title .....	381
2.3.101	box.confirm.dialog.show .....	381
	Erlaubte Werte .....	381
2.3.102	box.confirm.dialog.title .....	381
2.3.103	box.confirm.dialog.type .....	382
	Erlaubte Werte .....	382
2.3.104	box.confirm.dialog.content.....	382
2.3.105	box.confirm.dialog.footer.show.....	383
	Erlaubte Werte .....	383
2.3.106	box.confirm.dialog.footer.button.close.title.....	383
2.3.107	box.confirm.dialog.footer.button.cancel.show .....	384
	Erlaubte Werte .....	384
2.3.108	box.confirm.dialog.footer.button.cancel.title .....	384
2.3.109	dashlet.load.json.script.first .....	385
2.3.110	dashlet.json.path .....	385
2.3.111	chart.custom.init.function.....	386
2.3.112	bar.n[].component.n[].initbuttons .....	386
2.3.113	bar.n[].component.n[].onRemoveButton .....	387
2.3.114	bar.n[].component.n[].onAddButton .....	387
2.3.115	html.....	388
2.3.116	box.header.style.....	388

2.3.117	box.title.style.....	389
2.3.118	box.subtitle.style.....	389
2.3.119	box.content.style .....	389
2.3.120	box.block.style .....	390
2.3.121	box.headerbar.style .....	390
2.3.122	box.headerbar.html .....	391
2.3.123	chart.predefined.series.....	391
2.3.124	chart.predefined.legend.use.series .....	392
2.3.125	chart.custom.openpanel.function .....	392
2.3.126	responsive.grid.....	393
2.3.127	title.....	393
2.3.128	subtitle.....	394
2.3.129	chart.event.dataviewchanged .....	394
2.3.130	chart.event.axisareaselected .....	395
2.3.131	chart.event.brush.....	395
2.3.132	chart.event.brushselected.....	396
2.3.133	chart.event.datazoom .....	396
2.3.134	chart.event.datarangeselectd .....	397
2.3.135	chart.event.dbclick .....	397
2.3.136	chart.event.finished .....	398
2.3.137	chart.event.focusnodeadjacency.....	398
2.3.138	chart.event.globalcursortaken.....	399
2.3.139	chart.event.mousedown .....	399
2.3.140	chart.event.mousemove.....	400
2.3.141	chart.event.mouseout .....	401
2.3.142	chart.event.mouseover .....	401
2.3.143	chart.event.mouseup.....	402
2.3.144	chart.event.globalout .....	402
2.3.145	chart.event.legendselectchanged .....	403
2.3.146	chart.event.legendselected.....	403
2.3.147	chart.event.legendunselected .....	404
2.3.148	chart.event.legendselectall.....	404
2.3.149	chart.event.legendscroll.....	405
2.3.150	chart.event.unfocusnodeadjacency .....	406
2.3.151	chart.event.magictypechanged .....	406
2.3.152	chart.event.mapselected.....	407

2.3.153	chart.event.mapunselected .....	407
2.3.154	chart.event.mapselectchanged .....	408
2.3.155	chart.event.pieselectd .....	408
2.3.156	chart.event.pieunselected.....	409
2.3.157	chart.event.pieselectchanged.....	409
2.3.158	chart.event.rendered .....	410
2.3.159	chart.event.restore .....	410
2.3.160	chart.event.timelinechanged .....	411
2.3.161	chart.event.timelineplaychanged.....	411
2.3.162	chart.series.map.showLegendSymbol .....	412
2.3.163	chart.series.map.dataRangeHoverLink .....	412
2.3.164	chart.series.center .....	413
2.3.165	chart.series.zoom.....	413
2.3.166	chart.series.map.hoverable.....	414
2.3.167	chart.series.map.mapLocation.x .....	414
2.3.168	chart.series.map.mapLocation.y .....	415
2.3.169	chart.series.map.mapLocation.width .....	416
2.3.170	chart.series.map.mapLocation.height .....	416
2.3.171	locale .....	417
2.3.172	id .....	417
2.3.173	bar.n[].component.n[].select.type.....	418
2.3.174	box.download.event.link.....	418
2.3.175	box.download.link .....	419
2.3.176	exportChart.dashletId.....	419
2.3.177	exportChart.javascript .....	420
2.3.178	exportChart.dpi .....	420
2.3.179	chart.series.filter .....	421
2.3.180	box.expand.show .....	421
2.3.181	box.expand.hide.ids.....	422
2.3.182	modal.box.....	422
2.3.183	box.modal.size .....	422
	Erlaubte Werte .....	423
2.4	Dashlet Engine KPI .....	423
2.4.1	kpi.box.count .....	423
2.4.2	kpi.box.i[].type .....	424
2.4.3	kpi.box.i[].value.....	424

2.4.4	kpi.box.i[].valueseries .....	424
2.4.5	kpi.box.i[].trendvalue .....	425
2.4.6	kpi.box.i[].valueformat .....	425
2.4.7	kpi.box.i[].footer .....	425
2.4.8	kpi.box.i[].style .....	426
2.4.9	kpi.box.i[].progressbar.background.style .....	426
2.4.10	kpi.box.i[].progressbar.style.{x} .....	427
2.4.11	kpi.box.i[].progressbar.class.{x} .....	427
2.4.12	kpi.box.i[].progressbar.tooltip.{x}.show .....	428
2.4.13	kpi.box.i[].progressbar.tooltip.{x}.placement .....	428
2.4.14	kpi.box.i[].progressbar.tooltip.{x}.html .....	429
2.4.15	kpi.box.i[].progressbar.tooltip.{x}.title .....	429
2.4.16	kpi.box.i[].progressbar.name .....	429
2.4.17	kpi.box.i[].progressbar.name.{x} .....	430
2.4.18	kpi.box.i[].footer.visible .....	430
2.4.19	kpi.box.i[].footer.style .....	430
2.4.20	kpi.box.i[].footer.text.style .....	431
2.4.21	kpi.box.i[].title .....	431
2.4.22	kpi.box.i[].title.visible .....	431
2.4.23	kpi.box.i[].title.style .....	432
2.4.24	kpi.box.i[].title.text.style .....	432
2.4.25	kpi.box.i[].value.style .....	433
2.4.26	kpi.box.content.style .....	433
2.4.27	kpi.box.i[].value.onclick .....	434
2.4.28	kpi.box.i[].echarts.inline.id .....	434
2.4.29	kpi.box.i[].chart.height .....	435
2.4.30	kpi.box.i[].trend.chart.type .....	435
2.4.31	kpi.box.i[].trend.chart.width .....	436
2.4.32	kpi.box.i[].trend.chart.drawNormalOnTop .....	436
2.4.33	kpi.box.i[].trend.chart.valueSpots .....	437
2.4.34	kpi.box.i[].trend.chart.lineColor .....	437
2.4.35	kpi.box.i[].trend.chart.chartRangeMin .....	438
2.4.36	kpi.box.i[].trend.chart.chartRangeMax .....	438
2.4.37	kpi.box.i[].trend.chart.defaultPixelsPerValue .....	439
2.4.38	kpi.box.i[].trend.chart.spotColor .....	439
2.4.39	kpi.box.i[].trend.chart.minSpotColor .....	439

2.4.40	kpi.box.i[].trend.chart.maxSpotColor.....	440
2.4.41	kpi.box.i[].trend.chart.spotRadius.....	440
2.4.42	kpi.box.i[].trend.chart.highlightSpotColor.....	441
2.4.43	kpi.box.i[].trend.chart.highlightLineColor.....	441
2.4.44	kpi.box.i[].trend.chart.lineWidth.....	442
2.4.45	kpi.box.i[].trend.chart.normalRangeMin.....	442
2.4.46	kpi.box.i[].trend.chart.normalRangeMax.....	443
2.4.47	kpi.box.i[].trend.chart.chartRangeMaxX.....	443
2.4.48	kpi.box.i[].trend.chart.zeroAxis.....	443
2.4.49	kpi.box.i[].trend.chart.xvalues.....	444
2.4.50	kpi.box.i[].trend.chart.tooltipOffsetX.....	444
2.4.51	kpi.box.i[].trend.chart.barSpacing.....	445
2.4.52	kpi.box.i[].trend.chart.barColor.....	445
2.4.53	kpi.box.i[].trend.chart.negBarColor.....	446
2.4.54	kpi.box.i[].trend.chart.zeroColor.....	446
2.4.55	kpi.box.i[].trend.chart.nullColor.....	446
2.4.56	kpi.box.i[].trend.chart.colorMap.....	447
2.4.57	kpi.box.i[].trend.chart.stackedBarColor.....	447
2.4.58	kpi.box.i[].trend.chart.posBarColor.....	448
2.4.59	kpi.box.i[].trend.chart.zeroBarColor.....	448
2.4.60	kpi.box.i[].trend.chart.lineHeight.....	448
2.4.61	kpi.box.i[].trend.chart.thresholdValue.....	449
2.4.62	kpi.box.i[].trend.chart.thresholdColor.....	449
2.4.63	kpi.box.i[].trend.chart.targetColor.....	450
2.4.64	kpi.box.i[].trend.chart.targetWidth.....	450
2.4.65	kpi.box.i[].trend.chart.performanceColor.....	451
2.4.66	kpi.box.i[].trend.chart.sliceColors.....	451
2.4.67	kpi.box.i[].trend.chart.offset.....	451
2.4.68	kpi.box.i[].trend.chart.borderWidth.....	452
2.4.69	kpi.box.i[].trend.chart.borderColor.....	452
2.4.70	kpi.box.i[].trend.chart.raw.....	453
2.4.71	kpi.box.i[].trend.chart.showOutliers.....	453
2.4.72	kpi.box.i[].trend.chart.outlierIQR.....	454
2.4.73	kpi.box.i[].trend.chart.boxLineColor.....	454
2.4.74	kpi.box.i[].trend.chart.boxFillColor.....	455
2.4.75	kpi.box.i[].trend.chart.whiskerColor.....	455

2.4.76	kpi.box.i[].trend.chart.outlierFillColor.....	455
2.4.77	kpi.box.i[].trend.chart.minValue .....	456
2.4.78	kpi.box.i[].trend.chart.maxValue .....	456
2.4.79	kpi.box.i[].trend.chart.medianColor .....	457
2.4.80	kpi.box.i[].trend.chart.barSpacing.....	457
2.4.81	kpi.box.i[].trend.chart.disableInteraction .....	458
2.4.82	kpi.box.i[].trend.chart.disableTooltips.....	458
2.4.83	kpi.box.i[].trend.chart.highlightLighten .....	458
2.4.84	kpi.box.i[].trend.chart.tooltipOffsetY .....	459
2.4.85	kpi.box.i[].trend.chart.tooltipFormatter .....	459
2.4.86	kpi.box.i[].trend.chart.tooltipChartTitle.....	460
2.4.87	kpi.box.i[].trend.chart.tooltipFormat .....	460
2.4.88	kpi.box.i[].trend.chart.tooltipPrefix.....	460
2.4.89	kpi.box.i[].trend.chart.tooltipSuffix .....	461
2.4.90	kpi.box.i[].trend.chart.tooltipSkipNull .....	461
2.4.91	kpi.box.i[].trend.chart.tooltipValueLookups.....	461
2.4.92	kpi.box.i[].trend.chart.tooltipFormatFieldlist.....	462
2.4.93	kpi.box.i[].trend.chart.tooltipFormatFieldlist.....	462
2.4.94	kpi.box.i[].trend.chart.tooltipContainer.....	462
2.4.95	kpi.box.i[].trend.chart.tooltipClassname .....	463
2.4.96	kpi.box.i[].trend.chart.numberFormatter .....	463
2.4.97	kpi.box.i[].trend.chart.numberDigitGroupSep.....	464
2.4.98	kpi.box.i[].trend.chart.numberDecimalMark .....	464
2.4.99	kpi.box.i[].trend.chart.numberDigitGroupCount.....	464
2.4.100	kpi.box.i[].icon.up .....	465
2.4.101	kpi.box.i[].icon.down .....	465
2.4.102	kpi.box.i[].icon.down.style .....	466
2.4.103	kpi.box.i[].icon.up.style .....	466
2.4.104	kpi.box.i[].label.down.....	467
2.4.105	kpi.box.i[].label.up.....	467
2.4.106	kpi.box.i[].value.up.style .....	467
2.4.107	kpi.box.i[].value.down.style .....	468
2.4.108	kpi.box.i[].trendvalue.style .....	468
2.4.109	kpi.box.i[].trendvalueformat.....	469
2.4.110	kpi.box.responsive.....	469
2.4.111	kpi.box.i[].icon .....	470

2.4.112	kpi.box.i[].trend.chart.height .....	470
2.4.113	kpi.box.i[].trend.chart.fillColor .....	471
2.4.114	box.style .....	471
2.4.115	kpi.box.i[].trendvalue.visible.....	471
2.4.116	kpi.box.i[].icon.style.....	472
2.4.117	kpi.box.i[].icon.visible.....	472
<b>2.5</b>	<b>Dashlet Engine Special Dashlet.....</b>	<b>473</b>
2.5.1	parameter.layout .....	473
	Erlaubte Werte .....	473
2.5.2	parameter.toggleVisibility .....	473
2.5.3	parameter.toggleVisibility.defaultState .....	474
	Erlaubte Werte .....	474
2.5.4	parameter.showHiddenParameters .....	474
2.5.5	parameter.multiValueSeperator .....	475
	Erlaubte Werte .....	475
2.5.6	parameter.helpTextAsTooltip .....	475
2.5.7	parameter.showPasswordClearText .....	475
2.5.8	parameter.param.n[].show .....	476
2.5.9	parameter.param.n[].labelText.....	476
2.5.10	parameter.param.n[].helpText .....	477
2.5.11	parameter.showTextOutputParameters .....	477
<b>3</b>	<b>DataTables [Options] .....</b>	<b>478</b>
3.1	DataTables Buttons .....	478
3.1.1	table.buttons.buttons.....	478
3.1.2	Typ .....	479
	array.....	479
3.1.3	Beispiele .....	479
3.2	DataTables General.....	480
3.2.1	table.feature.autoWidth .....	480
3.2.2	Beispiel .....	480
3.2.3	Verwandt .....	480
	table.feature.deferRender .....	481
3.2.4	Beispiele .....	481
3.2.5	Verwandt .....	482
	table.feature.info .....	482



3.2.6	Beispiel .....	482
3.2.7	Verwandt .....	483
	table.feature.lengthChange .....	483
3.2.8	Beispiel .....	483
3.2.9	Verwandt .....	484
	table.feature.ordering .....	484
3.2.10	Beispiel .....	484
3.2.11	Verwandt .....	484
	table.feature.paging .....	485
3.2.12	Beispiel .....	485
3.2.13	Verwandt .....	485
	table.feature.processing.....	486
3.2.14	Beispiel .....	486
3.2.15	Verwandt .....	486
	table.feature.scrollX.....	486
3.2.16	Beispiel .....	487
3.2.17	Verwandt .....	487
	table.feature.scrollY.....	487
3.2.18	Beispiel .....	488
3.2.19	Verwandt .....	488
	table.feature.searching .....	488
3.2.20	Beispiel .....	489
3.2.21	Verwandt .....	489
	table.feature.serverSide .....	489
3.2.22	Beispiel .....	490
3.2.23	Verwandt .....	490
	table.feature.stateSave .....	490
3.2.24	Beispiel .....	491
3.2.25	Verwandt .....	491
	table.data .....	491
3.2.26	Beispiel .....	492
3.2.27	Verwandt .....	493
	table.callback.createdRow.....	493
3.2.28	Typ .....	494
	Funktion createdRow( row, data, dataIndex, cells ) .....	494
3.2.29	Beispiel .....	494

3.2.30	Verwandt .....	494
	table.callback.drawCallback .....	495
3.2.31	Typ .....	495
	Funktion drawCallback( settings ) .....	495
3.2.32	Beispiele .....	495
3.2.33	Verwandt .....	496
	table.callback.footerCallback .....	496
3.2.34	Typ .....	496
	Funktion footerCallback( tfoot, data, start, end, display ) .....	496
3.2.35	Beispiel .....	496
3.2.36	Verwandt .....	497
	table.callback.formatNumber .....	497
3.2.37	Type .....	498
	Funktion formatNumber( toFormat ) .....	498
3.2.38	Beispiel .....	498
3.2.39	Verwandt .....	498
	table.callback.headerCallback .....	498
3.2.40	Typ .....	499
	Funktion headerCallback( thead, data, start, end, display ) .....	499
3.2.41	Beispiel .....	499
3.2.42	Verwandt .....	499
	table.callback.infoCallback .....	500
3.2.43	Typ .....	500
	Funktion infoCallback( settings, start, end, max, total, pre ) .....	500
3.2.44	Beispiel .....	500
3.2.45	Verwandt .....	501
	table.callback.initComplete .....	501
3.2.46	Typ .....	502
	Funktion initComplete( settings, json ) .....	502
3.2.47	Beispiel .....	502
3.2.48	Verwandt .....	502
	table.callback.preDrawCallback .....	502
3.2.49	Typ .....	503
	Funktion preDrawCallback( settings ) .....	503
3.2.50	Beispiel .....	503
3.2.51	Verwandt .....	503

	table.callback.rowCallback .....	504
3.2.52	Typ .....	504
	Funktion rowCallback( row, data, displayNum, displayIndex, dataIndex ) .....	504
3.2.53	Beispiele .....	504
3.2.54	Verwandt .....	505
	table.callback.stateLoadCallback.....	505
3.2.55	Typ .....	506
	Funktion stateLoadCallback( settings, callback ) .....	506
3.2.56	Beispiel .....	506
3.2.57	Verwandt .....	507
	table.callback.stateLoaded .....	507
3.2.58	Typ .....	508
	Funktion stateLoaded( settings, data ) .....	508
3.2.59	Beispiel .....	508
3.2.60	Verwandt .....	508
	table.callback.stateLoadParams .....	509
3.2.61	Typ .....	509
	Funktion stateLoadParams( settings, data ) .....	509
3.2.62	Beispiel .....	509
3.2.63	Verwandt .....	510
	table.callback.stateSaveCallback.....	510
3.2.64	Typ .....	511
	Funktion stateSaveCallback( settings, data ) .....	511
3.2.65	Beispiel .....	511
3.2.66	Verwandt .....	512
	table.deferLoading.....	512
3.2.67	Typen .....	513
	integer.....	513
	array.....	513
3.2.68	Beispiel .....	513
3.2.69	Verwandt .....	514
	table.destroy .....	514
3.2.70	Beispiel .....	514
3.2.71	Verwandt .....	514
	table.displayStart.....	515
3.2.72	Beispiel .....	515

3.2.73	Verwandt .....	515
	table.dom .....	516
	Optionen.....	516
	Markup.....	516
	Styling.....	517
	Plug-ins.....	518
3.2.74	Beispiel .....	518
3.2.75	Verwandt .....	519
	table.lengthMenu .....	520
3.2.76	Beispiel .....	520
3.2.77	Verwandt .....	520
	table.order.....	521
3.2.78	Beispiele .....	521
3.2.79	Verwandt .....	522
	Erlaubte Werte .....	522
	table.orderCellsTop .....	522
3.2.80	Beispiel .....	523
3.2.81	Verwandt .....	523
	table.orderClasses .....	523
3.2.82	Beispiel .....	523
3.2.83	Verwandt .....	524
	table.orderFixed.....	524
3.2.84	Typen .....	525
	Array.....	525
	Objekt .....	525
3.2.85	Beispiele .....	525
3.2.86	Verwandt .....	526
	table.orderMulti .....	526
3.2.87	Beispiel .....	526
3.2.88	Verwandt .....	527
	table.pageLength.....	527
3.2.89	Beispiel .....	527
3.2.90	Verwandt .....	528
	table.pagingType .....	528
3.2.91	Beispiel .....	528
3.2.92	Verwandt .....	529

	Erlaubte Werte .....	529
	table.renderer .....	529
3.2.93	Typen .....	530
	string .....	530
	objekt.....	530
3.2.94	Beispiele .....	530
3.2.95	Verwandt .....	530
	table.retrieve .....	531
3.2.96	Beispiel .....	531
3.2.97	Verwandt .....	531
	table.rowId .....	532
3.2.98	Beispiel .....	532
3.2.99	Verwandt .....	533
	table.scrollCollapse .....	533
3.2.100	Beispiel .....	533
3.2.101	Verwandt .....	533
	table.search.caseInsensitive .....	534
3.2.102	Beispiel .....	534
3.2.103	Verwandt .....	534
	table.search.regex.....	534
3.2.104	Beispiel .....	535
3.2.105	Verwandt .....	535
	table.search.smart.....	535
3.2.106	Beispiel .....	536
3.2.107	Verwandt .....	536
	table.search.search.....	536
3.2.108	Beispiel .....	537
3.2.109	Verwandt .....	537
	table.searchCols.....	537
3.2.110	Beispiel .....	538
3.2.111	Verwandt .....	538
	table.searchDelay .....	538
3.2.112	Beschreibung .....	539
3.2.113	Beispiel .....	539
3.2.114	Verwandt .....	539
	table.stateDuration.....	539

3.2.115	Beispiele .....	540
3.2.116	Verwandt .....	540
	table.stripeClasses .....	541
3.2.117	Beispiel .....	541
	table.tabIndex .....	541
3.2.118	Beispiel .....	542
3.2.119	Verwandt .....	542
	table.columnDef.t[].cellType.....	542
3.2.120	Beispiel .....	542
3.2.121	Verwandt .....	543
	Erlaubte Werte .....	543
	table.columnDef.t[].className.....	543
3.2.122	Beispiel .....	543
	table.columnDef.t[].contentPadding.....	544
3.2.123	Beispiel .....	544
3.2.124	Verwandt .....	544
	table.columnDef.t[].data .....	544
3.2.125	Types.....	545
	integer.....	545
	string.....	545
	null .....	546
	objekt.....	546
	funktion data( row, type, set, meta ).....	546
3.2.126	Beispiele .....	547
3.2.127	Verwandt .....	549
	table.columnDef.t[].defaultContent .....	550
3.2.128	Beispiele .....	550
3.2.129	Verwandt .....	551
	table.columnDef.t[].name .....	551
3.2.130	Beispiel .....	552
3.2.131	Verwandt .....	552
	table.columnDef.t[].orderable .....	552
3.2.132	Beispiel .....	553
3.2.133	Verwandt .....	553
	table.columnDef.t[].orderData.....	553
3.2.134	Beispiel .....	554

3.2.135	Verwandt .....	554
	table.columnDef.t[].orderDataType .....	554
3.2.136	Beispiel .....	555
3.2.137	Verwandt .....	555
	table.columnDef.t[].orderSequence .....	555
3.2.138	Beispiel .....	555
3.2.139	Verwandt .....	556
	Erlaubte Werte .....	556
	table.columnDef.t[].render.....	556
	Array zugriff .....	557
3.2.140	Typen .....	557
	integer.....	557
	string.....	557
	objekt.....	558
	funktion render( data, type, row, meta ).....	558
3.2.141	Beispiele .....	559
3.2.142	Verwandt .....	562
	table.columnDef.t[].searchable .....	562
3.2.143	Beispiel .....	562
3.2.144	Verwandt .....	562
	table.columnDef.t[].title .....	563
3.2.145	Beispiel .....	563
	table.columnDef.t[].type .....	563
3.2.146	Beispiel .....	565
3.2.147	Verwandt .....	565
	table.columnDef.t[].visible.....	565
3.2.148	Beispiel .....	565
3.2.149	Verwandt .....	566
	table.columnDef.t[].width .....	566
3.2.150	Beispiel .....	566
3.2.151	Verwandt .....	567
	table.columns.i[].cellType.....	567
3.2.152	Beispiel .....	567
3.2.153	Verwandt .....	568
	Erlaubte Werte .....	568
	table.columns.i[].className.....	568

3.2.154	Beispiel .....	568
	table.columns.i[].contentPadding.....	569
3.2.155	Beispiel .....	569
3.2.156	Verwandt .....	569
	table.columns.i[].data .....	569
3.2.157	Types.....	570
	integer.....	570
	string.....	570
	null .....	571
	objekt.....	571
	funktion data( row, type, set, meta ).....	571
3.2.158	Beispiele .....	572
3.2.159	Verwandt .....	574
	table.columns.i[].defaultContent .....	575
3.2.160	Beispiele .....	575
3.2.161	Verwandt .....	576
	table.columns.i[].name .....	576
3.2.162	Beispiel .....	577
3.2.163	Verwandt .....	577
	table.columns.i[].orderable.....	577
3.2.164	Beispiel .....	578
3.2.165	Verwandt .....	578
	table.columns.i[].orderData .....	578
3.2.166	Beispiel .....	579
3.2.167	Verwandt .....	579
	table.columns.i[].orderDataType.....	579
3.2.168	Beispiel .....	580
3.2.169	Verwandt .....	580
	table.columns.i[].orderSequence .....	580
3.2.170	Beispiel .....	581
3.2.171	Verwandt .....	581
	Erlaubte Werte .....	581
	table.columns.i[].render.....	581
	Array zugriff .....	582
3.2.172	Typen .....	583
	integer.....	583



	string.....	583
	objekt.....	583
	funktion render( data, type, row, meta ).....	583
3.2.173	Beispiele .....	584
3.2.174	Verwandt .....	587
	table.columns.i[].searchable.....	587
3.2.175	Beispiel .....	587
3.2.176	Verwandt .....	587
	table.columns.i[].title .....	588
3.2.177	Beispiel .....	588
	table.columns.i[].type.....	588
3.2.178	Beispiel .....	590
3.2.179	Verwandt .....	590
	table.columns.i[].visible .....	590
3.2.180	Beispiel .....	591
3.2.181	Verwandt .....	591
	table.columns.i[].width .....	591
3.2.182	Beispiel .....	592
3.2.183	Verwandt .....	592
	table.language.aria.paginate.first.....	592
3.2.184	Beispiel .....	592
3.2.185	Verwandt .....	593
	table.language.aria.paginate.last .....	593
3.2.186	Beispiel .....	593
3.2.187	Verwandt .....	594
	table.language.aria.paginate.next.....	594
3.2.188	Beispiel .....	594
3.2.189	Verwandt .....	595
	table.language.aria.paginate.previous.....	595
3.2.190	Beispiel .....	595
3.2.191	Verwandt .....	596
	table.language.aria.paginate .....	596
3.2.192	Beispiel .....	596
3.2.193	Verwandt .....	597
	table.language.aria.sortAscending .....	597
3.2.194	Beispiel .....	598

3.2.195	Verwandt .....	598
	table.language.aria.sortDescending .....	598
3.2.196	Beispiel .....	598
3.2.197	Verwandt .....	599
	table.language.aria .....	599
3.2.198	Verwandt .....	599
	table.language.decimal .....	599
3.2.199	Beispiel .....	600
3.2.200	Verwandt .....	601
	table.language.emptyTable .....	601
3.2.201	Beispiel .....	601
3.2.202	Verwandt .....	601
	table.language.info .....	602
3.2.203	Beispiel .....	602
3.2.204	Verwandt .....	602
	table.language.infoEmpty .....	603
3.2.205	Beispiel .....	603
3.2.206	Verwandt .....	603
	table.language.infoFiltered .....	603
3.2.207	Beispiel .....	604
3.2.208	Verwandt .....	604
	table.language.infoPostFix.....	604
3.2.209	Beispiel .....	605
3.2.210	Verwandt .....	605
	table.language.lengthMenu .....	605
3.2.211	Beispiele .....	606
3.2.212	Verwandt .....	606
	table.language.loadingRecords .....	606
3.2.213	Beispiel .....	607
3.2.214	Verwandt .....	607
	table.language.paginate.first.....	607
3.2.215	Beispiel .....	608
3.2.216	Verwandt .....	608
	table.language.paginate.last .....	608
3.2.217	Beispiel .....	608
3.2.218	Verwandt .....	609

	table.language.paginate.next .....	609
3.2.219	Beispiel .....	609
3.2.220	Verwandt .....	610
	table.language.paginate.previous .....	610
3.2.221	Beispiel .....	610
3.2.222	Verwandt .....	611
	table.language.paginate.....	611
3.2.223	Verwandt .....	611
	table.language.processing .....	612
3.2.224	Beispiel .....	612
3.2.225	Verwandt .....	612
	table.language.search .....	612
3.2.226	Beispiele .....	613
3.2.227	Verwandt .....	613
	table.language.searchPlaceholder .....	613
3.2.228	Beispiele .....	614
3.2.229	Verwandt .....	614
	table.language.thousands.....	614
3.2.230	Beispiel .....	615
3.2.231	Verwandt .....	615
	table.language.url.....	615
3.2.232	Beispiele .....	616
3.2.233	Verwandt .....	617
	table.language.zeroRecords.....	617
3.2.234	Beispiel .....	617
3.2.235	Verwandt .....	617
	table.\$fn.dataTable.ext.errMode .....	618
3.2.236	Typen .....	618
	string .....	618
	Funktion \$.fn.dataTable.ext.errMode( settings, techNote, message ) .....	618
3.2.237	Beispiel .....	619
3.2.238	Verwandt .....	619
	Erlaubte Werte .....	619
	table.colReorder.fixedColumnsLeft.....	619
3.2.239	Beispiel .....	620
3.2.240	Verwandt .....	620

	table.colReorder.fixedColumnsRight .....	620
3.2.241	Beispiel .....	620
3.2.242	Verwandt .....	621
	table.colReorder.order .....	621
3.2.243	Typ .....	621
	array.....	621
3.2.244	Beispiel .....	621
3.2.245	Verwandt .....	622
	table.colReorder.realtime .....	622
3.2.246	Beispiel .....	622
	table.colReorder .....	623
3.2.247	Typen .....	623
	boolean.....	623
	objekt.....	623
3.2.248	Beispiele .....	623
	table.fixedColumns.heightMatch.....	624
3.2.249	Beispiel .....	624
	Erlaubte Werte .....	624
	table.fixedColumns.leftColumns .....	625
3.2.250	Beispiel .....	625
	table.fixedColumns.rightColumns .....	625
3.2.251	Beispiele .....	626
	table.fixedColumns.....	626
3.2.252	Typen .....	627
	boolean.....	627
	objekt.....	627
3.2.253	Beispiele .....	627
	table.fixedHeader.footer .....	627
3.2.254	Beispiel .....	628
	table.fixedHeader.footerOffset .....	628
3.2.255	Beispiele .....	629
3.2.256	Verwandt .....	629
	table.fixedHeader.header.....	629
3.2.257	Beispiel .....	630
	table.fixedHeader.headerOffset.....	630
3.2.258	Beispiele .....	630

3.2.259	Verwandt .....	631
	table.fixedHeader .....	631
3.2.260	Typen .....	632
	boolean.....	632
	objekt.....	632
3.2.261	Beispiele .....	632
	table.style .....	632
	table.language.locale .....	633
	table.columnDef.i[].className .....	633
3.2.262	Beispiel .....	633
	table.columnDef.n[].className.....	634
3.2.263	Beispiel .....	634
	table.colReorder.enable .....	634
3.2.264	Beispiel .....	635
	table.complex.header.....	635
3.2.265	Beispiel .....	635
	table.ext.search.....	635
	table.ext.search.minmaxColumnIndex.....	636
	table.ext.search.startSearchMode .....	637
	table.ext.search.searchModeList .....	638
	table.ext.search.searchModeButtonNames .....	639
	table.ext.search.searchModeButtonVisible.....	640
	table.ext.search.caseinsensitiveSearch.....	641
	table.ext.search.caseinsensitiveCheckBoxVisible.....	642
	table.ext.search.caseinsensitiveText.....	642
	table.ext.search.inputBarShow.....	643
	table.ext.search.inputBarMax .....	644
	table.ext.search.inputBarMin .....	645
	table.ext.search.inputBarMaxtext.....	646
	table.ext.search.inputBarMintext .....	646
	table.ext.search.minmaxColumnName .....	647
	table.ext.search.minmaxColumnType.....	648
	table.ext.search.hiddenSearch .....	649
3.3	DataTables Responsive.....	650
3.3.1	table.columns.i[].responsivePriority .....	650
3.3.2	Beispiele .....	650

	table.columnDef.t[].responsivePriority .....	651
3.3.3	Beispiele .....	651
	table.responsive.breakpoints .....	652
3.3.4	Typ .....	653
	array.....	653
3.3.5	Beispiele .....	653
	table.responsive.details .....	654
3.3.6	Typen .....	654
	boolean.....	654
	object.....	655
3.3.7	Beispiele .....	655
	table.responsive.details.display .....	655
3.3.8	Typ .....	656
	funktion display( row, update, render ) .....	656
3.3.9	Beispiele .....	656
	table.responsive.details.renderer.....	657
3.3.10	Typ .....	658
	funktion renderer( api, rowIdx, columns ) .....	658
3.3.11	Beispiele .....	659
	table.responsive.details.type.....	660
3.3.12	Typ .....	661
	string.....	661
3.3.13	Beispiel .....	661
	Erlaubte Werte .....	661
	table.responsive.details.target .....	661
3.3.14	Typen .....	662
	integer.....	662
	string.....	662
3.3.15	Beispiele .....	662
	table.responsive.orthogonal.....	663
3.3.16	Beispiel .....	663
	Erlaubte Werte .....	664
	table.responsive.....	664
3.3.17	Typen .....	664
	boolean.....	664
	objekt.....	665

3.3.18	Beispiele .....	665
<b>4</b>	<b>ECharts (2) [Options].....</b>	<b>666</b>
4.1	ECharts (2) Toolbox.....	666
4.1.1	chart.toolbox.feature.toggleSeriesVisibility.show .....	666
<b>5</b>	<b>ECharts (4) [Options].....</b>	<b>667</b>
5.1	ECharts (4) Angle Axis.....	667
5.1.1	chart.angleAxis.object .....	667
5.1.2	chart.angleAxis.id.....	667
5.1.3	chart.angleAxis.polarIndex.....	668
5.1.4	chart.angleAxis.startAngle.....	668
5.1.5	chart.angleAxis.clockwise .....	668
5.1.6	chart.angleAxis.type .....	669
	Erlaubte Werte .....	669
5.1.7	chart.angleAxis.boundaryGap.....	669
5.1.8	chart.angleAxis.min .....	670
5.1.9	chart.angleAxis.max.....	670
5.1.10	chart.angleAxis.scale .....	671
5.1.11	chart.angleAxis.splitNumber.....	671
5.1.12	chart.angleAxis.minInterval .....	672
5.1.13	chart.angleAxis.interval .....	672
5.1.14	chart.angleAxis.logBase.....	673
5.1.15	chart.angleAxis.silent.....	673
5.1.16	chart.angleAxis.triggerEvent .....	673
5.1.17	chart.angleAxis.axisLine.object.....	674
5.1.18	chart.angleAxis.axisLine.show .....	674
5.1.19	chart.angleAxis.axisLine.symbol .....	675
	Erlaubte Werte .....	675
5.1.20	chart.angleAxis.axisLine.symbolSize .....	675
5.1.21	chart.angleAxis.axisLine.symbolOffset .....	676
5.1.22	chart.angleAxis.axisLine.lineStyle.object .....	676
5.1.23	chart.angleAxis.axisLine.lineStyle.color .....	676
5.1.24	chart.angleAxis.axisLine.lineStyle.width .....	677
5.1.25	chart.angleAxis.axisLine.lineStyle.type .....	678
	Erlaubte Werte .....	678
5.1.26	chart.angleAxis.axisLine.lineStyle.shadowBlur .....	678

5.1.27	chart.angleAxis.axisLine.lineStyle.shadowColor .....	679
5.1.28	chart.angleAxis.axisLine.lineStyle.shadowOffsetX .....	679
5.1.29	chart.angleAxis.axisLine.lineStyle.shadowOffsetY .....	679
5.1.30	chart.angleAxis.axisLine.lineStyle.opacity .....	680
5.1.31	chart.angleAxis.axisTick.object.....	680
5.1.32	chart.angleAxis.axisTick.show .....	681
5.1.33	chart.angleAxis.axisTick.alignWithLabel .....	681
5.1.34	chart.angleAxis.axisTick.interval.....	681
5.1.35	chart.angleAxis.axisTick.inside .....	682
5.1.36	chart.angleAxis.axisTick.length.....	682
5.1.37	chart.angleAxis.axisTick.lineStyle.object .....	682
5.1.38	chart.angleAxis.axisTick.lineStyle.color .....	683
5.1.39	chart.angleAxis.axisTick.lineStyle.width .....	683
5.1.40	chart.angleAxis.axisTick.lineStyle.type .....	683
	Erlaubte Werte .....	684
5.1.41	chart.angleAxis.axisTick.lineStyle.shadowBlur.....	684
5.1.42	chart.angleAxis.axisTick.lineStyle.shadowColor.....	684
5.1.43	chart.angleAxis.axisTick.lineStyle.shadowOffsetX .....	685
5.1.44	chart.angleAxis.axisTick.lineStyle.shadowOffsetY.....	685
5.1.45	chart.angleAxis.axisTick.lineStyle.opacity .....	685
5.1.46	chart.angleAxis.axisLabel.object.....	686
5.1.47	chart.angleAxis.axisLabel.show .....	686
5.1.48	chart.angleAxis.axisLabel.interval .....	686
5.1.49	chart.angleAxis.axisLabel.inside .....	687
5.1.50	chart.angleAxis.axisLabel.margin .....	687
5.1.51	chart.angleAxis.axisLabel.formatter.....	687
5.1.52	chart.angleAxis.axisLabel.showMinLabel.....	688
5.1.53	chart.angleAxis.axisLabel.showMaxLabel .....	689
5.1.54	chart.angleAxis.axisLabel.color .....	689
5.1.55	chart.angleAxis.axisLabel.fontStyle.....	690
	Erlaubte Werte .....	690
5.1.56	chart.angleAxis.axisLabel.fontWeight .....	690
	Erlaubte Werte .....	690
5.1.57	chart.angleAxis.axisLabel.fontFamily.....	691
5.1.58	chart.angleAxis.axisLabel.fontSize .....	691
5.1.59	chart.angleAxis.axisLabel.align.....	691



	Erlaubte Werte .....	692
5.1.60	chart.angleAxis.axisLabel.verticalAlign .....	692
	Erlaubte Werte .....	693
5.1.61	chart.angleAxis.axisLabel.lineHeight.....	693
5.1.62	chart.angleAxis.axisLabel.backgroundColor.....	693
5.1.63	chart.angleAxis.axisLabel.borderColor.....	694
5.1.64	chart.angleAxis.axisLabel.borderWidth.....	694
5.1.65	chart.angleAxis.axisLabel.borderRadius .....	695
5.1.66	chart.angleAxis.axisLabel.padding .....	695
5.1.67	chart.angleAxis.axisLabel.shadowColor .....	696
5.1.68	chart.angleAxis.axisLabel.shadowBlur .....	696
5.1.69	chart.angleAxis.axisLabel.shadowOffsetX.....	696
5.1.70	chart.angleAxis.axisLabel.shadowOffsetY .....	697
5.1.71	chart.angleAxis.axisLabel.width .....	697
5.1.72	chart.angleAxis.axisLabel.height .....	698
5.1.73	chart.angleAxis.axisLabel.textBorderColor .....	698
5.1.74	chart.angleAxis.axisLabel.textBorderWidth .....	698
5.1.75	chart.angleAxis.axisLabel.textShadowColor .....	699
5.1.76	chart.angleAxis.axisLabel.textShadowBlur .....	699
5.1.77	chart.angleAxis.axisLabel.textShadowOffsetX.....	699
5.1.78	chart.angleAxis.axisLabel.textShadowOffsetY .....	700
5.1.79	chart.angleAxis.axisLabel.rich.object .....	700
5.1.80	chart.angleAxis.splitLine.object.....	701
5.1.81	chart.angleAxis.splitLine.show .....	702
5.1.82	chart.angleAxis.splitLine.interval.....	702
5.1.83	chart.angleAxis.splitLine.lineStyle.object .....	702
5.1.84	chart.angleAxis.splitLine.lineStyle.color .....	703
5.1.85	chart.angleAxis.splitLine.lineStyle.width .....	703
5.1.86	chart.angleAxis.splitLine.lineStyle.type .....	704
	Erlaubte Werte .....	704
5.1.87	chart.angleAxis.splitLine.lineStyle.shadowBlur.....	704
5.1.88	chart.angleAxis.splitLine.lineStyle.shadowColor.....	705
5.1.89	chart.angleAxis.splitLine.lineStyle.shadowOffsetX.....	705
5.1.90	chart.angleAxis.splitLine.lineStyle.shadowOffsetY.....	705
5.1.91	chart.angleAxis.splitLine.lineStyle.opacity .....	706
5.1.92	chart.angleAxis.splitArea.object .....	706

5.1.93	chart.angleAxis.splitArea.interval .....	706
5.1.94	chart.angleAxis.splitArea.show .....	707
5.1.95	chart.angleAxis.splitArea.areaStyle.object.....	707
5.1.96	chart.angleAxis.splitArea.areaStyle.color .....	707
5.1.97	chart.angleAxis.splitArea.areaStyle.shadowBlur .....	708
5.1.98	chart.angleAxis.splitArea.areaStyle.shadowColor .....	708
5.1.99	chart.angleAxis.splitArea.areaStyle.shadowOffsetX .....	709
5.1.100	chart.angleAxis.splitArea.areaStyle.shadowOffsetY .....	709
5.1.101	chart.angleAxis.splitArea.areaStyle.opacity.....	709
5.1.102	chart.angleAxis.axisPointer.object .....	710
5.1.103	chart.angleAxis.axisPointer.show .....	710
5.1.104	chart.angleAxis.axisPointer.type .....	710
	Erlaubte Werte .....	711
5.1.105	chart.angleAxis.axisPointer.snap.....	711
5.1.106	chart.angleAxis.axisPointer.z .....	711
5.1.107	chart.angleAxis.axisPointer.label.object .....	712
5.1.108	chart.angleAxis.axisPointer.label.show.....	712
5.1.109	chart.angleAxis.axisPointer.label.precision .....	712
5.1.110	chart.angleAxis.axisPointer.label.formatter .....	713
5.1.111	chart.angleAxis.axisPointer.label.margin.....	714
5.1.112	chart.angleAxis.axisPointer.label.color .....	715
5.1.113	chart.angleAxis.axisPointer.label.fontStyle .....	715
	Erlaubte Werte .....	715
5.1.114	chart.angleAxis.axisPointer.label.fontWeight .....	716
	Erlaubte Werte .....	716
5.1.115	chart.angleAxis.axisPointer.label.fontFamily.....	716
5.1.116	chart.angleAxis.axisPointer.label.fontSize .....	717
5.1.117	chart.angleAxis.axisPointer.label.lineHeight .....	717
5.1.118	chart.angleAxis.axisPointer.label.width .....	718
5.1.119	chart.angleAxis.axisPointer.label.height.....	718
5.1.120	chart.angleAxis.axisPointer.label.textBorderColor.....	718
5.1.121	chart.angleAxis.axisPointer.label.textBorderWidth.....	719
5.1.122	chart.angleAxis.axisPointer.label.textShadowColor .....	719
5.1.123	chart.angleAxis.axisPointer.label.textShadowBlur.....	720
5.1.124	chart.angleAxis.axisPointer.label.textShadowOffsetX .....	720
5.1.125	chart.angleAxis.axisPointer.label.textShadowOffsetY.....	720

5.1.126	chart.angleAxis.axisPointer.label.padding.....	721
5.1.127	chart.angleAxis.axisPointer.label.backgroundColor .....	721
5.1.128	chart.angleAxis.axisPointer.label.borderColor .....	722
5.1.129	chart.angleAxis.axisPointer.label.borderWidth .....	722
5.1.130	chart.angleAxis.axisPointer.label.shadowBlur.....	722
5.1.131	chart.angleAxis.axisPointer.label.shadowColor .....	723
5.1.132	chart.angleAxis.axisPointer.label.shadowOffsetX .....	723
5.1.133	chart.angleAxis.axisPointer.label.shadowOffsetY.....	724
5.1.134	chart.angleAxis.axisPointer.lineStyle.object.....	724
5.1.135	chart.angleAxis.axisPointer.lineStyle.color.....	724
5.1.136	chart.angleAxis.axisPointer.lineStyle.width.....	726
5.1.137	chart.angleAxis.axisPointer.lineStyle.type.....	726
	Erlaubte Werte .....	726
5.1.138	chart.angleAxis.axisPointer.lineStyle.shadowBlur .....	726
5.1.139	chart.angleAxis.axisPointer.lineStyle.shadowColor .....	727
5.1.140	chart.angleAxis.axisPointer.lineStyle.shadowOffsetX .....	727
5.1.141	chart.angleAxis.axisPointer.lineStyle.shadowOffsetY .....	728
5.1.142	chart.angleAxis.axisPointer.lineStyle.opacity.....	728
5.1.143	chart.angleAxis.axisPointer.shadowStyle.object.....	728
5.1.144	chart.angleAxis.axisPointer.shadowStyle.color.....	729
5.1.145	chart.angleAxis.axisPointer.shadowStyle.shadowBlur .....	730
5.1.146	chart.angleAxis.axisPointer.shadowStyle.shadowColor .....	731
5.1.147	chart.angleAxis.axisPointer.shadowStyle.shadowOffsetX .....	731
5.1.148	chart.angleAxis.axisPointer.shadowStyle.shadowOffsetY .....	732
5.1.149	chart.angleAxis.axisPointer.shadowStyle.opacity.....	732
5.1.150	chart.angleAxis.axisPointer.triggerTooltip.....	732
5.1.151	chart.angleAxis.axisPointer.value.....	733
5.1.152	chart.angleAxis.axisPointer.status.....	733
	Erlaubte Werte .....	733
5.1.153	chart.angleAxis.axisPointer.handle.object.....	733
5.1.154	chart.angleAxis.axisPointer.handle.show .....	734
5.1.155	chart.angleAxis.axisPointer.handle.icon .....	734
5.1.156	chart.angleAxis.axisPointer.handle.size .....	735
5.1.157	chart.angleAxis.axisPointer.handle.margin .....	735
5.1.158	chart.angleAxis.axisPointer.handle.color.....	735
5.1.159	chart.angleAxis.axisPointer.handle.throttle .....	736

5.1.160	chart.angleAxis.axisPointer.handle.shadowBlur .....	736
5.1.161	chart.angleAxis.axisPointer.handle.shadowColor .....	737
5.1.162	chart.angleAxis.axisPointer.handle.shadowOffsetX .....	737
5.1.163	chart.angleAxis.axisPointer.handle.shadowOffsetY .....	737
5.1.164	chart.angleAxis.zlevel .....	738
5.1.165	chart.angleAxis.z .....	738
5.2	ECharts (4) Axis.....	739
5.2.1	chart.yAxis.{x}.object .....	739
5.2.2	chart.yAxis.{x}.id .....	739
5.2.3	chart.yAxis.{x}.show .....	739
5.2.4	chart.yAxis.{x}.gridIndex .....	740
5.2.5	chart.yAxis.{x}.position .....	740
	Erlaubte Werte .....	741
5.2.6	chart.yAxis.{x}.offset .....	741
5.2.7	chart.yAxis.{x}.type.....	741
	Erlaubte Werte .....	742
5.2.8	chart.yAxis.{x}.name .....	742
5.2.9	chart.yAxis.{x}.nameLocation.....	742
	Erlaubte Werte .....	742
5.2.10	chart.yAxis.{x}.nameTextStyle.object .....	743
5.2.11	chart.yAxis.{x}.nameTextStyle.color .....	743
5.2.12	chart.yAxis.{x}.nameTextStyle.fontStyle .....	743
	Erlaubte Werte .....	744
5.2.13	chart.yAxis.{x}.nameTextStyle.fontWeight .....	744
	Erlaubte Werte .....	744
5.2.14	chart.yAxis.{x}.nameTextStyle.fontFamily.....	744
5.2.15	chart.yAxis.{x}.nameTextStyle.fontSize .....	745
5.2.16	chart.yAxis.{x}.nameTextStyle.align .....	745
	Erlaubte Werte .....	746
5.2.17	chart.yAxis.{x}.nameTextStyle.verticalAlign.....	746
	Erlaubte Werte .....	746
5.2.18	chart.yAxis.{x}.nameTextStyle.lineHeight .....	747
5.2.19	chart.yAxis.{x}.nameTextStyle.backgroundColor .....	747
5.2.20	chart.yAxis.{x}.nameTextStyle.borderColor .....	748
5.2.21	chart.yAxis.{x}.nameTextStyle.borderWidth .....	748
5.2.22	chart.yAxis.{x}.nameTextStyle.borderRadius.....	748

5.2.23	chart.yAxis.{x}.nameTextStyle.padding .....	749
5.2.24	chart.yAxis.{x}.nameTextStyle.shadowColor .....	749
5.2.25	chart.yAxis.{x}.nameTextStyle.shadowBlur .....	750
5.2.26	chart.yAxis.{x}.nameTextStyle.shadowOffsetX .....	750
5.2.27	chart.yAxis.{x}.nameTextStyle.shadowOffsetY .....	750
5.2.28	chart.yAxis.{x}.nameTextStyle.width .....	751
5.2.29	chart.yAxis.{x}.nameTextStyle.height .....	751
5.2.30	chart.yAxis.{x}.nameTextStyle.textBorderColor .....	752
5.2.31	chart.yAxis.{x}.nameTextStyle.textBorderWidth .....	752
5.2.32	chart.yAxis.{x}.nameTextStyle.textShadowColor .....	752
5.2.33	chart.yAxis.{x}.nameTextStyle.textShadowBlur .....	753
5.2.34	chart.yAxis.{x}.nameTextStyle.textShadowOffsetX .....	753
5.2.35	chart.yAxis.{x}.nameTextStyle.textShadowOffsetY .....	753
5.2.36	chart.yAxis.{x}.nameTextStyle.rich.object .....	754
5.2.37	chart.yAxis.{x}.nameGap .....	755
5.2.38	chart.yAxis.{x}.nameRotate .....	756
5.2.39	chart.yAxis.{x}.inverse .....	756
5.2.40	chart.yAxis.{x}.boundaryGap .....	756
5.2.41	chart.yAxis.{x}.min .....	757
5.2.42	chart.yAxis.{x}.max .....	757
5.2.43	chart.yAxis.{x}.scale .....	758
5.2.44	chart.yAxis.{x}.splitNumber .....	758
5.2.45	chart.yAxis.{x}.minInterval .....	759
5.2.46	chart.yAxis.{x}.interval .....	759
5.2.47	chart.yAxis.{x}.logBase .....	760
5.2.48	chart.yAxis.{x}.silent .....	760
5.2.49	chart.yAxis.{x}.triggerEvent .....	760
5.2.50	chart.yAxis.{x}.axisLine.object .....	761
5.2.51	chart.yAxis.{x}.axisLine.show .....	761
5.2.52	chart.yAxis.{x}.axisLine.onZero .....	762
5.2.53	chart.yAxis.{x}.axisLine.onZeroAxisIndex .....	762
5.2.54	chart.yAxis.{x}.axisLine.symbol .....	762
	Erlaubte Werte .....	763
5.2.55	chart.yAxis.{x}.axisLine.symbolSize .....	763
5.2.56	chart.yAxis.{x}.axisLine.symbolOffset .....	763
5.2.57	chart.yAxis.{x}.axisLine.lineStyle.object .....	764

5.2.58	chart.yAxis.{x}.axisLine.lineStyle.color .....	764
5.2.59	chart.yAxis.{x}.axisLine.lineStyle.width .....	765
5.2.60	chart.yAxis.{x}.axisLine.lineStyle.type .....	766
	Erlaubte Werte .....	766
5.2.61	chart.yAxis.{x}.axisLine.lineStyle.shadowBlur .....	766
5.2.62	chart.yAxis.{x}.axisLine.lineStyle.shadowColor .....	767
5.2.63	chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetX .....	767
5.2.64	chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetY .....	767
5.2.65	chart.yAxis.{x}.axisLine.lineStyle.opacity .....	768
5.2.66	chart.yAxis.{x}.axisTick.object .....	768
5.2.67	chart.yAxis.{x}.axisTick.show .....	768
5.2.68	chart.yAxis.{x}.axisTick.alignWithLabel .....	769
5.2.69	chart.yAxis.{x}.axisTick.interval .....	769
5.2.70	chart.yAxis.{x}.axisTick.inside .....	770
5.2.71	chart.yAxis.{x}.axisTick.length .....	770
5.2.72	chart.yAxis.{x}.axisTick.lineStyle.object .....	770
5.2.73	chart.yAxis.{x}.axisTick.lineStyle.color .....	771
5.2.74	chart.yAxis.{x}.axisTick.lineStyle.width .....	771
5.2.75	chart.yAxis.{x}.axisTick.lineStyle.type .....	771
	Erlaubte Werte .....	772
5.2.76	chart.yAxis.{x}.axisTick.lineStyle.shadowBlur .....	772
5.2.77	chart.yAxis.{x}.axisTick.lineStyle.shadowColor .....	772
5.2.78	chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetX .....	773
5.2.79	chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetY .....	773
5.2.80	chart.yAxis.{x}.axisTick.lineStyle.opacity .....	773
5.2.81	chart.yAxis.{x}.axisLabel.object .....	774
5.2.82	chart.yAxis.{x}.axisLabel.show .....	774
5.2.83	chart.yAxis.{x}.axisLabel.interval .....	774
5.2.84	chart.yAxis.{x}.axisLabel.inside .....	775
5.2.85	chart.yAxis.{x}.axisLabel.rotate .....	775
5.2.86	chart.yAxis.{x}.axisLabel.margin .....	775
5.2.87	chart.yAxis.{x}.axisLabel.formatter .....	776
5.2.88	chart.yAxis.{x}.axisLabel.showMinLabel .....	776
5.2.89	chart.yAxis.{x}.axisLabel.showMaxLabel .....	777
5.2.90	chart.yAxis.{x}.axisLabel.color .....	777
5.2.91	chart.yAxis.{x}.axisLabel.fontStyle .....	778

	Erlaubte Werte .....	778
5.2.92	chart.yAxis.{x}.axisLabel.fontWeight.....	778
	Erlaubte Werte .....	779
5.2.93	chart.yAxis.{x}.axisLabel.fontFamily .....	779
5.2.94	chart.yAxis.{x}.axisLabel.fontSize.....	779
5.2.95	chart.yAxis.{x}.axisLabel.align .....	780
	Erlaubte Werte .....	780
5.2.96	chart.yAxis.{x}.axisLabel.verticalAlign .....	780
	Erlaubte Werte .....	781
5.2.97	chart.yAxis.{x}.axisLabel.lineHeight.....	781
5.2.98	chart.yAxis.{x}.axisLabel.backgroundColor .....	782
5.2.99	chart.yAxis.{x}.axisLabel.borderColor.....	782
5.2.100	chart.yAxis.{x}.axisLabel.borderWidth .....	783
5.2.101	chart.yAxis.{x}.axisLabel.borderRadius .....	783
5.2.102	chart.yAxis.{x}.axisLabel.padding .....	783
5.2.103	chart.yAxis.{x}.axisLabel.shadowColor .....	784
5.2.104	chart.yAxis.{x}.axisLabel.shadowBlur .....	784
5.2.105	chart.yAxis.{x}.axisLabel.shadowOffsetX.....	785
5.2.106	chart.yAxis.{x}.axisLabel.shadowOffsetY .....	785
5.2.107	chart.yAxis.{x}.axisLabel.width.....	785
5.2.108	chart.yAxis.{x}.axisLabel.height.....	786
5.2.109	chart.yAxis.{x}.axisLabel.textBorderColor .....	786
5.2.110	chart.yAxis.{x}.axisLabel.textBorderWidth .....	787
5.2.111	chart.yAxis.{x}.axisLabel.textShadowColor .....	787
5.2.112	chart.yAxis.{x}.axisLabel.textShadowBlur .....	787
5.2.113	chart.yAxis.{x}.axisLabel.textShadowOffsetX.....	788
5.2.114	chart.yAxis.{x}.axisLabel.textShadowOffsetY .....	788
5.2.115	chart.yAxis.{x}.axisLabel.rich.object .....	788
5.2.116	chart.yAxis.{x}.splitLine.object .....	789
5.2.117	chart.yAxis.{x}.splitLine.show.....	790
5.2.118	chart.yAxis.{x}.splitLine.interval.....	790
5.2.119	chart.yAxis.{x}.splitLine.lineStyle.object .....	790
5.2.120	chart.yAxis.{x}.splitLine.lineStyle.color .....	791
5.2.121	chart.yAxis.{x}.splitLine.lineStyle.width .....	791
5.2.122	chart.yAxis.{x}.splitLine.lineStyle.type.....	792
	Erlaubte Werte .....	792

5.2.123	chart.yAxis.{x}.splitLine.lineStyle.shadowBlur .....	792
5.2.124	chart.yAxis.{x}.splitLine.lineStyle.shadowColor .....	793
5.2.125	chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetX .....	793
5.2.126	chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetY .....	793
5.2.127	chart.yAxis.{x}.splitLine.lineStyle.opacity .....	794
5.2.128	chart.yAxis.{x}.splitArea.object .....	794
5.2.129	chart.yAxis.{x}.splitArea.interval .....	794
5.2.130	chart.yAxis.{x}.splitArea.show .....	795
5.2.131	chart.yAxis.{x}.splitArea.areaStyle.object .....	795
5.2.132	chart.yAxis.{x}.splitArea.areaStyle.color .....	795
5.2.133	chart.yAxis.{x}.splitArea.areaStyle.shadowBlur .....	796
5.2.134	chart.yAxis.{x}.splitArea.areaStyle.shadowColor .....	796
5.2.135	chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetX .....	797
5.2.136	chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetY .....	797
5.2.137	chart.yAxis.{x}.splitArea.areaStyle.opacity .....	797
5.2.138	chart.yAxis.{x}.axisPointer.object .....	798
5.2.139	chart.yAxis.{x}.axisPointer.show .....	798
5.2.140	chart.yAxis.{x}.axisPointer.type .....	799
	Erlaubte Werte .....	799
5.2.141	chart.yAxis.{x}.axisPointer.snap .....	799
5.2.142	chart.yAxis.{x}.axisPointer.z .....	799
5.2.143	chart.yAxis.{x}.axisPointer.label.object .....	800
5.2.144	chart.yAxis.{x}.axisPointer.label.show .....	800
5.2.145	chart.yAxis.{x}.axisPointer.label.precision .....	801
5.2.146	chart.yAxis.{x}.axisPointer.label.formatter .....	801
5.2.147	chart.yAxis.{x}.axisPointer.label.margin .....	802
5.2.148	chart.yAxis.{x}.axisPointer.label.color .....	803
5.2.149	chart.yAxis.{x}.axisPointer.label.fontStyle .....	803
	Erlaubte Werte .....	803
5.2.150	chart.yAxis.{x}.axisPointer.label.fontWeight .....	804
	Erlaubte Werte .....	804
5.2.151	chart.yAxis.{x}.axisPointer.label.fontFamily .....	804
5.2.152	chart.yAxis.{x}.axisPointer.label.fontSize .....	805
5.2.153	chart.yAxis.{x}.axisPointer.label.lineHeight .....	805
5.2.154	chart.yAxis.{x}.axisPointer.label.width .....	806
5.2.155	chart.yAxis.{x}.axisPointer.label.height .....	806



5.2.156	chart.yAxis.{x}.axisPointer.label.textBorderColor.....	806
5.2.157	chart.yAxis.{x}.axisPointer.label.textBorderWidth.....	807
5.2.158	chart.yAxis.{x}.axisPointer.label.textShadowColor.....	807
5.2.159	chart.yAxis.{x}.axisPointer.label.textShadowBlur.....	808
5.2.160	chart.yAxis.{x}.axisPointer.label.textShadowOffsetX.....	808
5.2.161	chart.yAxis.{x}.axisPointer.label.textShadowOffsetY.....	808
5.2.162	chart.yAxis.{x}.axisPointer.label.padding.....	809
5.2.163	chart.yAxis.{x}.axisPointer.label.backgroundColor.....	809
5.2.164	chart.yAxis.{x}.axisPointer.label.borderColor.....	810
5.2.165	chart.yAxis.{x}.axisPointer.label.borderWidth.....	810
5.2.166	chart.yAxis.{x}.axisPointer.label.shadowBlur.....	810
5.2.167	chart.yAxis.{x}.axisPointer.label.shadowColor.....	811
5.2.168	chart.yAxis.{x}.axisPointer.label.shadowOffsetX.....	811
5.2.169	chart.yAxis.{x}.axisPointer.label.shadowOffsetY.....	812
5.2.170	chart.yAxis.{x}.axisPointer.lineStyle.object.....	812
5.2.171	chart.yAxis.{x}.axisPointer.lineStyle.color.....	812
5.2.172	chart.yAxis.{x}.axisPointer.lineStyle.width.....	814
5.2.173	chart.yAxis.{x}.axisPointer.lineStyle.type.....	814
	Erlaubte Werte.....	814
5.2.174	chart.yAxis.{x}.axisPointer.lineStyle.shadowBlur.....	814
5.2.175	chart.yAxis.{x}.axisPointer.lineStyle.shadowColor.....	815
5.2.176	chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetX.....	815
5.2.177	chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetY.....	816
5.2.178	chart.yAxis.{x}.axisPointer.lineStyle.opacity.....	816
5.2.179	chart.yAxis.{x}.axisPointer.shadowStyle.object.....	816
5.2.180	chart.yAxis.{x}.axisPointer.shadowStyle.color.....	817
5.2.181	chart.yAxis.{x}.axisPointer.shadowStyle.shadowBlur.....	818
5.2.182	chart.yAxis.{x}.axisPointer.shadowStyle.shadowColor.....	819
5.2.183	chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetX.....	819
5.2.184	chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetY.....	820
5.2.185	chart.yAxis.{x}.axisPointer.shadowStyle.opacity.....	820
5.2.186	chart.yAxis.{x}.axisPointer.triggerTooltip.....	820
5.2.187	chart.yAxis.{x}.axisPointer.value.....	821
5.2.188	chart.yAxis.{x}.axisPointer.status.....	821
	Erlaubte Werte.....	821
5.2.189	chart.yAxis.{x}.axisPointer.handle.object.....	821

5.2.190	chart.yAxis.{x}.axisPointer.handle.show.....	822
5.2.191	chart.yAxis.{x}.axisPointer.handle.icon .....	822
5.2.192	chart.yAxis.{x}.axisPointer.handle.size .....	823
5.2.193	chart.yAxis.{x}.axisPointer.handle.margin .....	823
5.2.194	chart.yAxis.{x}.axisPointer.handle.color .....	823
5.2.195	chart.yAxis.{x}.axisPointer.handle.throttle.....	824
5.2.196	chart.yAxis.{x}.axisPointer.handle.shadowBlur .....	824
5.2.197	chart.yAxis.{x}.axisPointer.handle.shadowColor .....	825
5.2.198	chart.yAxis.{x}.axisPointer.handle.shadowOffsetX .....	825
5.2.199	chart.yAxis.{x}.axisPointer.handle.shadowOffsetY .....	825
5.2.200	chart.yAxis.{x}.zlevel .....	826
5.2.201	chart.yAxis.{x}.z .....	826
5.2.202	chart.xAxis.{x}.object .....	827
5.2.203	chart.xAxis.{x}.id .....	827
5.2.204	chart.xAxis.{x}.show .....	827
5.2.205	chart.xAxis.{x}.gridIndex .....	828
5.2.206	chart.xAxis.{x}.position .....	828
	Erlaubte Werte .....	828
5.2.207	chart.xAxis.{x}.offset .....	829
5.2.208	chart.xAxis.{x}.type.....	829
	Erlaubte Werte .....	829
5.2.209	chart.xAxis.{x}.name.....	830
5.2.210	chart.xAxis.{x}.nameLocation.....	830
	Erlaubte Werte .....	830
5.2.211	chart.xAxis.{x}.nameTextStyle.object .....	830
5.2.212	chart.xAxis.{x}.nameTextStyle.color .....	831
5.2.213	chart.xAxis.{x}.nameTextStyle.fontStyle.....	831
	Erlaubte Werte .....	831
5.2.214	chart.xAxis.{x}.nameTextStyle.fontWeight .....	832
	Erlaubte Werte .....	832
5.2.215	chart.xAxis.{x}.nameTextStyle.fontFamily.....	832
5.2.216	chart.xAxis.{x}.nameTextStyle.fontSize .....	833
5.2.217	chart.xAxis.{x}.nameTextStyle.align.....	833
	Erlaubte Werte .....	833
5.2.218	chart.xAxis.{x}.nameTextStyle.verticalAlign .....	834
	Erlaubte Werte .....	834

5.2.219	chart.xAxis.{x}.nameTextStyle.lineHeight .....	834
5.2.220	chart.xAxis.{x}.nameTextStyle.backgroundColor .....	835
5.2.221	chart.xAxis.{x}.nameTextStyle.borderColor .....	835
5.2.222	chart.xAxis.{x}.nameTextStyle.borderWidth .....	836
5.2.223	chart.xAxis.{x}.nameTextStyle.borderRadius .....	836
5.2.224	chart.xAxis.{x}.nameTextStyle.padding .....	837
5.2.225	chart.xAxis.{x}.nameTextStyle.shadowColor .....	837
5.2.226	chart.xAxis.{x}.nameTextStyle.shadowBlur .....	837
5.2.227	chart.xAxis.{x}.nameTextStyle.shadowOffsetX .....	838
5.2.228	chart.xAxis.{x}.nameTextStyle.shadowOffsetY .....	838
5.2.229	chart.xAxis.{x}.nameTextStyle.width .....	838
5.2.230	chart.xAxis.{x}.nameTextStyle.height .....	839
5.2.231	chart.xAxis.{x}.nameTextStyle.textBorderColor .....	839
5.2.232	chart.xAxis.{x}.nameTextStyle.textBorderWidth .....	840
5.2.233	chart.xAxis.{x}.nameTextStyle.textShadowColor .....	840
5.2.234	chart.xAxis.{x}.nameTextStyle.textShadowBlur .....	840
5.2.235	chart.xAxis.{x}.nameTextStyle.textShadowOffsetX .....	841
5.2.236	chart.xAxis.{x}.nameTextStyle.textShadowOffsetY .....	841
5.2.237	chart.xAxis.{x}.nameTextStyle.rich.object .....	842
5.2.238	chart.xAxis.{x}.nameGap .....	843
5.2.239	chart.xAxis.{x}.nameRotate .....	843
5.2.240	chart.xAxis.{x}.inverse .....	843
5.2.241	chart.xAxis.{x}.boundaryGap .....	844
5.2.242	chart.xAxis.{x}.min .....	844
5.2.243	chart.xAxis.{x}.max .....	845
5.2.244	chart.xAxis.{x}.scale .....	845
5.2.245	chart.xAxis.{x}.splitNumber .....	846
5.2.246	chart.xAxis.{x}.minInterval .....	846
5.2.247	chart.xAxis.{x}.interval .....	846
5.2.248	chart.xAxis.{x}.logBase .....	847
5.2.249	chart.xAxis.{x}.silent .....	847
5.2.250	chart.xAxis.{x}.triggerEvent .....	848
5.2.251	chart.xAxis.{x}.axisLine.object .....	848
5.2.252	chart.xAxis.{x}.axisLine.show .....	849
5.2.253	chart.xAxis.{x}.axisLine.onZero .....	849
5.2.254	chart.xAxis.{x}.axisLine.onZeroAxisIndex .....	849

5.2.255	chart.xAxis.{x}.axisLine.symbol .....	850
	Erlaubte Werte .....	850
5.2.256	chart.xAxis.{x}.axisLine.symbolSize .....	850
5.2.257	chart.xAxis.{x}.axisLine.symbolOffset .....	851
5.2.258	chart.xAxis.{x}.axisLine.lineStyle.object .....	851
5.2.259	chart.xAxis.{x}.axisLine.lineStyle.color .....	851
5.2.260	chart.xAxis.{x}.axisLine.lineStyle.width .....	853
5.2.261	chart.xAxis.{x}.axisLine.lineStyle.type .....	853
	Erlaubte Werte .....	853
5.2.262	chart.xAxis.{x}.axisLine.lineStyle.shadowBlur .....	853
5.2.263	chart.xAxis.{x}.axisLine.lineStyle.shadowColor .....	854
5.2.264	chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetX .....	854
5.2.265	chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetY .....	855
5.2.266	chart.xAxis.{x}.axisLine.lineStyle.opacity .....	855
5.2.267	chart.xAxis.{x}.axisTick.object .....	855
5.2.268	chart.xAxis.{x}.axisTick.show .....	856
5.2.269	chart.xAxis.{x}.axisTick.alignWithLabel .....	856
5.2.270	chart.xAxis.{x}.axisTick.interval .....	856
5.2.271	chart.xAxis.{x}.axisTick.inside .....	857
5.2.272	chart.xAxis.{x}.axisTick.length .....	857
5.2.273	chart.xAxis.{x}.axisTick.lineStyle.object .....	857
5.2.274	chart.xAxis.{x}.axisTick.lineStyle.color .....	858
5.2.275	chart.xAxis.{x}.axisTick.lineStyle.width .....	858
5.2.276	chart.xAxis.{x}.axisTick.lineStyle.type .....	858
	Erlaubte Werte .....	859
5.2.277	chart.xAxis.{x}.axisTick.lineStyle.shadowBlur .....	859
5.2.278	chart.xAxis.{x}.axisTick.lineStyle.shadowColor .....	859
5.2.279	chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetX .....	860
5.2.280	chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetY .....	860
5.2.281	chart.xAxis.{x}.axisTick.lineStyle.opacity .....	860
5.2.282	chart.xAxis.{x}.axisLabel.object .....	861
5.2.283	chart.xAxis.{x}.axisLabel.show .....	861
5.2.284	chart.xAxis.{x}.axisLabel.interval .....	862
5.2.285	chart.xAxis.{x}.axisLabel.inside .....	862
5.2.286	chart.xAxis.{x}.axisLabel.rotate .....	862
5.2.287	chart.xAxis.{x}.axisLabel.margin .....	863

5.2.288	chart.xAxis.{x}.axisLabel.formatter .....	863
5.2.289	chart.xAxis.{x}.axisLabel.showMinLabel .....	864
5.2.290	chart.xAxis.{x}.axisLabel.showMaxLabel .....	864
5.2.291	chart.xAxis.{x}.axisLabel.color .....	865
5.2.292	chart.xAxis.{x}.axisLabel.fontStyle .....	865
	Erlaubte Werte .....	866
5.2.293	chart.xAxis.{x}.axisLabel.fontWeight .....	866
	Erlaubte Werte .....	866
5.2.294	chart.xAxis.{x}.axisLabel.fontFamily .....	866
5.2.295	chart.xAxis.{x}.axisLabel.fontSize .....	867
5.2.296	chart.xAxis.{x}.axisLabel.align .....	867
	Erlaubte Werte .....	868
5.2.297	chart.xAxis.{x}.axisLabel.verticalAlign .....	868
	Erlaubte Werte .....	868
5.2.298	chart.xAxis.{x}.axisLabel.lineHeight .....	869
5.2.299	chart.xAxis.{x}.axisLabel.backgroundColor .....	869
5.2.300	chart.xAxis.{x}.axisLabel.borderColor .....	870
5.2.301	chart.xAxis.{x}.axisLabel.borderWidth .....	870
5.2.302	chart.xAxis.{x}.axisLabel.borderRadius .....	871
5.2.303	chart.xAxis.{x}.axisLabel.padding .....	871
5.2.304	chart.xAxis.{x}.axisLabel.shadowColor .....	871
5.2.305	chart.xAxis.{x}.axisLabel.shadowBlur .....	872
5.2.306	chart.xAxis.{x}.axisLabel.shadowOffsetX .....	872
5.2.307	chart.xAxis.{x}.axisLabel.shadowOffsetY .....	872
5.2.308	chart.xAxis.{x}.axisLabel.width .....	873
5.2.309	chart.xAxis.{x}.axisLabel.height .....	873
5.2.310	chart.xAxis.{x}.axisLabel.textBorderColor .....	874
5.2.311	chart.xAxis.{x}.axisLabel.textBorderWidth .....	874
5.2.312	chart.xAxis.{x}.axisLabel.textShadowColor .....	874
5.2.313	chart.xAxis.{x}.axisLabel.textShadowBlur .....	875
5.2.314	chart.xAxis.{x}.axisLabel.textShadowOffsetX .....	875
5.2.315	chart.xAxis.{x}.axisLabel.textShadowOffsetY .....	876
5.2.316	chart.xAxis.{x}.axisLabel.rich.object .....	876
5.2.317	chart.xAxis.{x}.splitLine.object .....	877
5.2.318	chart.xAxis.{x}.splitLine.show .....	878
5.2.319	chart.xAxis.{x}.splitLine.interval .....	878

5.2.320	chart.xAxis.{x}.splitLine.lineStyle.object .....	878
5.2.321	chart.xAxis.{x}.splitLine.lineStyle.color .....	879
5.2.322	chart.xAxis.{x}.splitLine.lineStyle.width .....	879
5.2.323	chart.xAxis.{x}.splitLine.lineStyle.type.....	880
	Erlaubte Werte .....	880
5.2.324	chart.xAxis.{x}.splitLine.lineStyle.shadowBlur .....	880
5.2.325	chart.xAxis.{x}.splitLine.lineStyle.shadowColor .....	881
5.2.326	chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetX.....	881
5.2.327	chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetY .....	881
5.2.328	chart.xAxis.{x}.splitLine.lineStyle.opacity.....	882
5.2.329	chart.xAxis.{x}.splitArea.object.....	882
5.2.330	chart.xAxis.{x}.splitArea.interval .....	882
5.2.331	chart.xAxis.{x}.splitArea.show .....	883
5.2.332	chart.xAxis.{x}.splitArea.areaStyle.object.....	883
5.2.333	chart.xAxis.{x}.splitArea.areaStyle.color.....	883
5.2.334	chart.xAxis.{x}.splitArea.areaStyle.shadowBlur .....	884
5.2.335	chart.xAxis.{x}.splitArea.areaStyle.shadowColor .....	884
5.2.336	chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetX .....	885
5.2.337	chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetY .....	885
5.2.338	chart.xAxis.{x}.splitArea.areaStyle.opacity .....	885
5.2.339	chart.xAxis.{x}.axisPointer.object.....	886
5.2.340	chart.xAxis.{x}.axisPointer.show .....	886
5.2.341	chart.xAxis.{x}.axisPointer.type.....	886
	Erlaubte Werte .....	887
5.2.342	chart.xAxis.{x}.axisPointer.snap .....	887
5.2.343	chart.xAxis.{x}.axisPointer.z .....	887
5.2.344	chart.xAxis.{x}.axisPointer.label.object .....	888
5.2.345	chart.xAxis.{x}.axisPointer.label.show .....	888
5.2.346	chart.xAxis.{x}.axisPointer.label.precision .....	888
5.2.347	chart.xAxis.{x}.axisPointer.label.formatter .....	889
5.2.348	chart.xAxis.{x}.axisPointer.label.margin .....	890
5.2.349	chart.xAxis.{x}.axisPointer.label.color .....	891
5.2.350	chart.xAxis.{x}.axisPointer.label.fontStyle.....	891
	Erlaubte Werte .....	891
5.2.351	chart.xAxis.{x}.axisPointer.label.fontWeight .....	892
	Erlaubte Werte .....	892

5.2.352	chart.xAxis.{x}.axisPointer.label.fontFamily.....	892
5.2.353	chart.xAxis.{x}.axisPointer.label.fontSize .....	893
5.2.354	chart.xAxis.{x}.axisPointer.label.lineHeight.....	893
5.2.355	chart.xAxis.{x}.axisPointer.label.width .....	894
5.2.356	chart.xAxis.{x}.axisPointer.label.height .....	894
5.2.357	chart.xAxis.{x}.axisPointer.label.textBorderColor.....	894
5.2.358	chart.xAxis.{x}.axisPointer.label.textBorderWidth .....	895
5.2.359	chart.xAxis.{x}.axisPointer.label.textShadowColor.....	895
5.2.360	chart.xAxis.{x}.axisPointer.label.textShadowBlur.....	896
5.2.361	chart.xAxis.{x}.axisPointer.label.textShadowOffsetX.....	896
5.2.362	chart.xAxis.{x}.axisPointer.label.textShadowOffsetY.....	896
5.2.363	chart.xAxis.{x}.axisPointer.label.padding .....	897
5.2.364	chart.xAxis.{x}.axisPointer.label.backgroundColor.....	897
5.2.365	chart.xAxis.{x}.axisPointer.label.borderColor .....	898
5.2.366	chart.xAxis.{x}.axisPointer.label.borderWidth.....	898
5.2.367	chart.xAxis.{x}.axisPointer.label.shadowBlur.....	898
5.2.368	chart.xAxis.{x}.axisPointer.label.shadowColor.....	899
5.2.369	chart.xAxis.{x}.axisPointer.label.shadowOffsetX.....	899
5.2.370	chart.xAxis.{x}.axisPointer.label.shadowOffsetY.....	900
5.2.371	chart.xAxis.{x}.axisPointer.lineStyle.object .....	900
5.2.372	chart.xAxis.{x}.axisPointer.lineStyle.color .....	900
5.2.373	chart.xAxis.{x}.axisPointer.lineStyle.width .....	902
5.2.374	chart.xAxis.{x}.axisPointer.lineStyle.type .....	902
	Erlaubte Werte .....	902
5.2.375	chart.xAxis.{x}.axisPointer.lineStyle.shadowBlur .....	902
5.2.376	chart.xAxis.{x}.axisPointer.lineStyle.shadowColor .....	903
5.2.377	chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetX .....	903
5.2.378	chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetY .....	904
5.2.379	chart.xAxis.{x}.axisPointer.lineStyle.opacity .....	904
5.2.380	chart.xAxis.{x}.axisPointer.shadowStyle.object .....	904
5.2.381	chart.xAxis.{x}.axisPointer.shadowStyle.color .....	905
5.2.382	chart.xAxis.{x}.axisPointer.shadowStyle.shadowBlur .....	906
5.2.383	chart.xAxis.{x}.axisPointer.shadowStyle.shadowColor .....	907
5.2.384	chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetX .....	907
5.2.385	chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetY .....	908
5.2.386	chart.xAxis.{x}.axisPointer.shadowStyle.opacity .....	908

5.2.387	chart.xAxis.{x}.axisPointer.triggerTooltip .....	908
5.2.388	chart.xAxis.{x}.axisPointer.value .....	909
5.2.389	chart.xAxis.{x}.axisPointer.status .....	909
	Erlaubte Werte .....	909
5.2.390	chart.xAxis.{x}.axisPointer.handle.object .....	909
5.2.391	chart.xAxis.{x}.axisPointer.handle.show .....	910
5.2.392	chart.xAxis.{x}.axisPointer.handle.icon .....	910
5.2.393	chart.xAxis.{x}.axisPointer.handle.size .....	911
5.2.394	chart.xAxis.{x}.axisPointer.handle.margin.....	911
5.2.395	chart.xAxis.{x}.axisPointer.handle.color .....	911
5.2.396	chart.xAxis.{x}.axisPointer.handle.throttle.....	912
5.2.397	chart.xAxis.{x}.axisPointer.handle.shadowBlur .....	912
5.2.398	chart.xAxis.{x}.axisPointer.handle.shadowColor .....	913
5.2.399	chart.xAxis.{x}.axisPointer.handle.shadowOffsetX .....	913
5.2.400	chart.xAxis.{x}.axisPointer.handle.shadowOffsetY .....	913
5.2.401	chart.xAxis.{x}.zlevel .....	914
5.2.402	chart.xAxis.{x}.z .....	914
5.3	<b>ECharts (4) Axis Pointer.....</b>	<b>915</b>
5.3.1	chart.axisPointer.object .....	915
5.3.2	chart.axisPointer.id.....	916
5.3.3	chart.axisPointer.show .....	916
5.3.4	chart.axisPointer.type .....	917
	Erlaubte Werte .....	917
5.3.5	chart.axisPointer.snap.....	917
5.3.6	chart.axisPointer.z .....	918
5.3.7	chart.axisPointer.label.object .....	918
5.3.8	chart.axisPointer.label.show.....	918
5.3.9	chart.axisPointer.label.precision .....	919
5.3.10	chart.axisPointer.label.formatter .....	919
5.3.11	chart.axisPointer.label.margin.....	920
5.3.12	chart.axisPointer.label.color .....	921
5.3.13	chart.axisPointer.label.fontStyle .....	921
	Erlaubte Werte .....	922
5.3.14	chart.axisPointer.label.fontWeight.....	922
	Erlaubte Werte .....	922
5.3.15	chart.axisPointer.label.fontFamily .....	922



5.3.16	chart.axisPointer.label.fontSize .....	923
5.3.17	chart.axisPointer.label.lineHeight .....	923
5.3.18	chart.axisPointer.label.width .....	924
5.3.19	chart.axisPointer.label.height .....	924
5.3.20	chart.axisPointer.label.textBorderColor .....	925
5.3.21	chart.axisPointer.label.textBorderWidth .....	925
5.3.22	chart.axisPointer.label.textShadowColor .....	925
5.3.23	chart.axisPointer.label.textShadowBlur .....	926
5.3.24	chart.axisPointer.label.textShadowOffsetX .....	926
5.3.25	chart.axisPointer.label.textShadowOffsetY .....	926
5.3.26	chart.axisPointer.label.padding .....	927
5.3.27	chart.axisPointer.label.backgroundColor .....	928
5.3.28	chart.axisPointer.label.borderColor .....	928
5.3.29	chart.axisPointer.label.borderWidth .....	928
5.3.30	chart.axisPointer.label.shadowBlur .....	929
5.3.31	chart.axisPointer.label.shadowColor .....	929
5.3.32	chart.axisPointer.label.shadowOffsetX .....	930
5.3.33	chart.axisPointer.label.shadowOffsetY .....	930
5.3.34	chart.axisPointer.lineStyle.object .....	930
5.3.35	chart.axisPointer.lineStyle.color .....	931
5.3.36	chart.axisPointer.lineStyle.width .....	932
5.3.37	chart.axisPointer.lineStyle.type .....	933
	Erlaubte Werte .....	933
5.3.38	chart.axisPointer.lineStyle.shadowBlur .....	933
5.3.39	chart.axisPointer.lineStyle.shadowColor .....	934
5.3.40	chart.axisPointer.lineStyle.shadowOffsetX .....	934
5.3.41	chart.axisPointer.lineStyle.shadowOffsetY .....	934
5.3.42	chart.axisPointer.lineStyle.opacity .....	935
5.3.43	chart.axisPointer.shadowStyle.object .....	935
5.3.44	chart.axisPointer.shadowStyle.color .....	935
5.3.45	chart.axisPointer.shadowStyle.shadowBlur .....	937
5.3.46	chart.axisPointer.shadowStyle.shadowColor .....	938
5.3.47	chart.axisPointer.shadowStyle.shadowOffsetX .....	938
5.3.48	chart.axisPointer.shadowStyle.shadowOffsetY .....	939
5.3.49	chart.axisPointer.shadowStyle.opacity .....	939
5.3.50	chart.axisPointer.triggerTooltip .....	939

5.3.51	chart.axisPointer.value .....	940
5.3.52	chart.axisPointer.status .....	940
	Erlaubte Werte .....	940
5.3.53	chart.axisPointer.handle.object .....	940
5.3.54	chart.axisPointer.handle.show .....	941
5.3.55	chart.axisPointer.handle.icon .....	941
5.3.56	chart.axisPointer.handle.size .....	942
5.3.57	chart.axisPointer.handle.margin .....	942
5.3.58	chart.axisPointer.handle.color .....	942
5.3.59	chart.axisPointer.handle.throttle .....	943
5.3.60	chart.axisPointer.handle.shadowBlur .....	943
5.3.61	chart.axisPointer.handle.shadowColor .....	944
5.3.62	chart.axisPointer.handle.shadowOffsetX .....	944
5.3.63	chart.axisPointer.handle.shadowOffsetY .....	944
5.3.64	chart.axisPointer.link .....	945
5.3.65	chart.axisPointer.triggerOn .....	946
	Erlaubte Werte .....	947
5.4	ECharts (4) Brush .....	947
5.4.1	chart.brush.object .....	947
5.4.2	chart.brush.id .....	947
5.4.3	chart.brush.toolbox .....	948
	Erlaubte Werte .....	948
5.4.4	chart.brush.brushLink .....	948
5.4.5	chart.brush.seriesIndex .....	949
5.4.6	chart.brush.geoIndex .....	950
5.4.7	chart.brush.xAxisIndex .....	951
5.4.8	chart.brush.yAxisIndex .....	952
5.4.9	chart.brush.brushType .....	953
	Erlaubte Werte .....	954
5.4.10	chart.brush.brushMode .....	954
	Erlaubte Werte .....	954
5.4.11	chart.brush.transformable .....	954
5.4.12	chart.brush.brushStyle .....	955
5.4.13	chart.brush.throttleType .....	955
5.4.14	chart.brush.throttleDelay .....	956
5.4.15	chart.brush.removeOnClick .....	956

5.4.16	chart.brush.inBrush .....	957
5.4.17	chart.brush.outOfBrush.....	957
5.4.18	chart.brush.z .....	958
5.5	<b>ECharts (4) Calendar .....</b>	<b>958</b>
5.5.1	chart.calendar.object .....	958
5.5.2	chart.calendar.id.....	959
5.5.3	chart.calendar.zlevel .....	959
5.5.4	chart.calendar.z .....	960
5.5.5	chart.calendar.left .....	960
	Erlaubte Werte .....	961
5.5.6	chart.calendar.top .....	961
	Erlaubte Werte .....	961
5.5.7	chart.calendar.right .....	961
	Erlaubte Werte .....	962
5.5.8	chart.calendar.bottom .....	962
	Erlaubte Werte .....	962
5.5.9	chart.calendar.width .....	962
5.5.10	chart.calendar.height .....	963
5.5.11	chart.calendar.range .....	963
5.5.12	chart.calendar.cellSize .....	964
5.5.13	chart.calendar.orient.....	965
	Erlaubte Werte .....	965
5.5.14	chart.calendar.splitLine.object .....	965
5.5.15	chart.calendar.splitLine.show.....	965
5.5.16	chart.calendar.splitLine.lineStyle.object .....	966
5.5.17	chart.calendar.splitLine.lineStyle.color .....	966
5.5.18	chart.calendar.splitLine.lineStyle.width .....	967
5.5.19	chart.calendar.splitLine.lineStyle.type.....	968
	Erlaubte Werte .....	968
5.5.20	chart.calendar.splitLine.lineStyle.shadowBlur .....	968
5.5.21	chart.calendar.splitLine.lineStyle.shadowColor.....	969
5.5.22	chart.calendar.splitLine.lineStyle.shadowOffsetX.....	969
5.5.23	chart.calendar.splitLine.lineStyle.shadowOffsetY .....	969
5.5.24	chart.calendar.splitLine.lineStyle.opacity.....	970
5.5.25	chart.calendar.itemStyle.object .....	970
5.5.26	chart.calendar.itemStyle.color .....	970

5.5.27	chart.calendar.itemStyle.borderColor .....	972
5.5.28	chart.calendar.itemStyle.borderWidth.....	973
5.5.29	chart.calendar.itemStyle.borderType .....	973
	Erlaubte Werte .....	973
5.5.30	chart.calendar.itemStyle.shadowBlur .....	974
5.5.31	chart.calendar.itemStyle.shadowColor.....	974
5.5.32	chart.calendar.itemStyle.shadowOffsetX.....	974
5.5.33	chart.calendar.itemStyle.shadowOffsetY .....	975
5.5.34	chart.calendar.itemStyle.opacity .....	975
5.5.35	chart.calendar.dayLabel.object .....	975
5.5.36	chart.calendar.dayLabel.show.....	976
5.5.37	chart.calendar.dayLabel.firstDay.....	976
5.5.38	chart.calendar.dayLabel.margin.....	977
5.5.39	chart.calendar.dayLabel.position.....	977
	Erlaubte Werte .....	977
5.5.40	chart.calendar.dayLabel.nameMap .....	978
5.5.41	chart.calendar.dayLabel.color .....	978
5.5.42	chart.calendar.dayLabel.fontStyle .....	979
	Erlaubte Werte .....	979
5.5.43	chart.calendar.dayLabel.fontWeight .....	979
	Erlaubte Werte .....	980
5.5.44	chart.calendar.dayLabel.fontFamily.....	980
5.5.45	chart.calendar.dayLabel.fontSize.....	980
5.5.46	chart.calendar.dayLabel.align .....	980
	Erlaubte Werte .....	981
5.5.47	chart.calendar.dayLabel.verticalAlign.....	981
	Erlaubte Werte .....	982
5.5.48	chart.calendar.dayLabel.lineHeight .....	982
5.5.49	chart.calendar.dayLabel.backgroundColor .....	982
5.5.50	chart.calendar.dayLabel.borderColor .....	983
5.5.51	chart.calendar.dayLabel.borderWidth .....	983
5.5.52	chart.calendar.dayLabel.borderRadius.....	984
5.5.53	chart.calendar.dayLabel.padding.....	984
5.5.54	chart.calendar.dayLabel.shadowColor .....	985
5.5.55	chart.calendar.dayLabel.shadowBlur.....	985
5.5.56	chart.calendar.dayLabel.shadowOffsetX .....	985

5.5.57	chart.calendar.dayLabel.shadowOffsetY.....	986
5.5.58	chart.calendar.dayLabel.width.....	986
5.5.59	chart.calendar.dayLabel.height.....	987
5.5.60	chart.calendar.dayLabel.textBorderColor.....	987
5.5.61	chart.calendar.dayLabel.textBorderWidth.....	987
5.5.62	chart.calendar.dayLabel.textShadowColor.....	988
5.5.63	chart.calendar.dayLabel.textShadowBlur.....	988
5.5.64	chart.calendar.dayLabel.textShadowOffsetX.....	988
5.5.65	chart.calendar.dayLabel.textShadowOffsetY.....	989
5.5.66	chart.calendar.dayLabel.rich.object.....	989
5.5.67	chart.calendar.monthLabel.object.....	990
5.5.68	chart.calendar.monthLabel.show.....	991
5.5.69	chart.calendar.monthLabel.align.....	991
	Erlaubte Werte.....	991
5.5.70	chart.calendar.monthLabel.margin.....	992
5.5.71	chart.calendar.monthLabel.position.....	992
	Erlaubte Werte.....	992
5.5.72	chart.calendar.monthLabel.nameMap.....	992
5.5.73	chart.calendar.monthLabel.formatter.....	993
5.5.74	chart.calendar.monthLabel.color.....	994
5.5.75	chart.calendar.monthLabel.fontStyle.....	995
	Erlaubte Werte.....	995
5.5.76	chart.calendar.monthLabel.fontWeight.....	995
	Erlaubte Werte.....	996
5.5.77	chart.calendar.monthLabel.fontFamily.....	996
5.5.78	chart.calendar.monthLabel.fontSize.....	996
5.5.79	chart.calendar.monthLabel.verticalAlign.....	996
	Erlaubte Werte.....	997
5.5.80	chart.calendar.monthLabel.lineHeight.....	997
5.5.81	chart.calendar.monthLabel.backgroundColor.....	998
5.5.82	chart.calendar.monthLabel.borderColor.....	998
5.5.83	chart.calendar.monthLabel.borderWidth.....	999
5.5.84	chart.calendar.monthLabel.borderRadius.....	999
5.5.85	chart.calendar.monthLabel.padding.....	999
5.5.86	chart.calendar.monthLabel.shadowColor.....	1000
5.5.87	chart.calendar.monthLabel.shadowBlur.....	1000

5.5.88	chart.calendar.monthLabel.shadowOffsetX .....	1001
5.5.89	chart.calendar.monthLabel.shadowOffsetY .....	1001
5.5.90	chart.calendar.monthLabel.width .....	1001
5.5.91	chart.calendar.monthLabel.height .....	1002
5.5.92	chart.calendar.monthLabel.textBorderColor .....	1002
5.5.93	chart.calendar.monthLabel.textBorderWidth .....	1003
5.5.94	chart.calendar.monthLabel.textShadowColor .....	1003
5.5.95	chart.calendar.monthLabel.textShadowBlur .....	1003
5.5.96	chart.calendar.monthLabel.textShadowOffsetX .....	1004
5.5.97	chart.calendar.monthLabel.textShadowOffsetY .....	1004
5.5.98	chart.calendar.monthLabel.rich.object .....	1004
5.5.99	chart.calendar.yearLabel.object .....	1005
5.5.100	chart.calendar.yearLabel.show .....	1006
5.5.101	chart.calendar.yearLabel.margin .....	1006
5.5.102	chart.calendar.yearLabel.position .....	1006
	Erlaubte Werte .....	1007
5.5.103	chart.calendar.yearLabel.formatter .....	1007
5.5.104	chart.calendar.yearLabel.color .....	1008
5.5.105	chart.calendar.yearLabel.fontStyle .....	1008
	Erlaubte Werte .....	1009
5.5.106	chart.calendar.yearLabel.fontWeight .....	1009
	Erlaubte Werte .....	1009
5.5.107	chart.calendar.yearLabel.fontFamily .....	1009
5.5.108	chart.calendar.yearLabel.fontSize .....	1010
5.5.109	chart.calendar.yearLabel.align .....	1010
	Erlaubte Werte .....	1011
5.5.110	chart.calendar.yearLabel.verticalAlign .....	1011
	Erlaubte Werte .....	1011
5.5.111	chart.calendar.yearLabel.lineHeight .....	1012
5.5.112	chart.calendar.yearLabel.backgroundColor .....	1012
5.5.113	chart.calendar.yearLabel.borderColor .....	1013
5.5.114	chart.calendar.yearLabel.borderWidth .....	1013
5.5.115	chart.calendar.yearLabel.borderRadius .....	1014
5.5.116	chart.calendar.yearLabel.padding .....	1014
5.5.117	chart.calendar.yearLabel.shadowColor .....	1014
5.5.118	chart.calendar.yearLabel.shadowBlur .....	1015

5.5.119	chart.calendar.yearLabel.shadowOffsetX .....	1015
5.5.120	chart.calendar.yearLabel.shadowOffsetY .....	1015
5.5.121	chart.calendar.yearLabel.width .....	1016
5.5.122	chart.calendar.yearLabel.height .....	1016
5.5.123	chart.calendar.yearLabel.textBorderColor .....	1017
5.5.124	chart.calendar.yearLabel.textBorderWidth .....	1017
5.5.125	chart.calendar.yearLabel.textShadowColor .....	1017
5.5.126	chart.calendar.yearLabel.textShadowBlur .....	1018
5.5.127	chart.calendar.yearLabel.textShadowOffsetX .....	1018
5.5.128	chart.calendar.yearLabel.textShadowOffsetY .....	1019
5.5.129	chart.calendar.yearLabel.rich.object .....	1019
5.5.130	chart.calendar.silent .....	1020
5.6	<b>ECharts (4) Data .....</b>	<b>1021</b>
5.6.1	chart.data.hierachy.render .....	1021
	Erlaubte Werte .....	1021
5.6.2	chart.dynamicSum.show .....	1021
5.6.3	chart.dynamicSum.position .....	1022
5.6.4	chart.dynamicSum.color .....	1022
5.6.5	chart.dynamicSum.roundedDigits .....	1022
5.6.6	chart.dynamicSum.excludeSeriesList .....	1023
5.6.7	chart.dynamicSum.retainStyleSeriesList .....	1023
5.6.8	chart.dynamicSum.retainStyleSeriesList .....	1024
5.6.9	chart.dynamicSum.customSeriesHandlerBlock .....	1024
5.6.10	chart.dynamicSum.customHandleLegendSelectChangedBlock .....	1025
5.7	<b>ECharts (4) Data Zoom .....</b>	<b>1025</b>
5.7.1	chart.dataZoom.object .....	1025
5.7.2	chart.dataZoom.type .....	1026
	Erlaubte Werte .....	1026
5.7.3	chart.dataZoom.id .....	1026
5.7.4	chart.dataZoom.show .....	1026
5.7.5	chart.dataZoom.backgroundColor .....	1027
5.7.6	chart.dataZoom.dataBackground.object .....	1027
5.7.7	chart.dataZoom.dataBackground.lineStyle.object .....	1027
5.7.8	chart.dataZoom.dataBackground.lineStyle.color .....	1028
5.7.9	chart.dataZoom.dataBackground.lineStyle.width .....	1029

5.7.10	chart.dataZoom.dataBackground.lineStyle.type .....	1030
	Erlaubte Werte .....	1030
5.7.11	chart.dataZoom.dataBackground.lineStyle.shadowBlur .....	1030
5.7.12	chart.dataZoom.dataBackground.lineStyle.shadowColor .....	1031
5.7.13	chart.dataZoom.dataBackground.lineStyle.shadowOffsetX .....	1031
5.7.14	chart.dataZoom.dataBackground.lineStyle.shadowOffsetY .....	1031
5.7.15	chart.dataZoom.dataBackground.lineStyle.opacity .....	1032
5.7.16	chart.dataZoom.dataBackground.areaStyle.object .....	1032
5.7.17	chart.dataZoom.dataBackground.areaStyle.color .....	1032
5.7.18	chart.dataZoom.dataBackground.areaStyle.shadowBlur .....	1034
5.7.19	chart.dataZoom.dataBackground.areaStyle.shadowColor .....	1035
5.7.20	chart.dataZoom.dataBackground.areaStyle.shadowOffsetX .....	1035
5.7.21	chart.dataZoom.dataBackground.areaStyle.shadowOffsetY .....	1036
5.7.22	chart.dataZoom.dataBackground.areaStyle.opacity .....	1036
5.7.23	chart.dataZoom.fillerColor .....	1036
5.7.24	chart.dataZoom.borderColor .....	1037
5.7.25	chart.dataZoom.handleIcon .....	1037
5.7.26	chart.dataZoom.handleSize .....	1038
	Erlaubte Werte .....	1038
5.7.27	chart.dataZoom.handleStyle.object .....	1038
5.7.28	chart.dataZoom.handleStyle.color .....	1038
5.7.29	chart.dataZoom.handleStyle.borderColor .....	1040
5.7.30	chart.dataZoom.handleStyle.borderWidth .....	1040
5.7.31	chart.dataZoom.handleStyle.borderType .....	1040
	Erlaubte Werte .....	1041
5.7.32	chart.dataZoom.handleStyle.shadowBlur .....	1041
5.7.33	chart.dataZoom.handleStyle.shadowColor .....	1041
5.7.34	chart.dataZoom.handleStyle.shadowOffsetX .....	1042
5.7.35	chart.dataZoom.handleStyle.shadowOffsetY .....	1042
5.7.36	chart.dataZoom.handleStyle.opacity .....	1042
5.7.37	chart.dataZoom.labelPrecision .....	1043
5.7.38	chart.dataZoom.labelFormatter .....	1043
5.7.39	chart.dataZoom.showDetail .....	1044
5.7.40	chart.dataZoom.showDataShadow .....	1044
5.7.41	chart.dataZoom.realtime .....	1045
5.7.42	chart.dataZoom.textStyle.object .....	1045



5.7.43	chart.dataZoom.textStyle.color .....	1045
5.7.44	chart.dataZoom.textStyle.fontStyle .....	1046
	Erlaubte Werte .....	1046
5.7.45	chart.dataZoom.textStyle.fontWeight .....	1046
	Erlaubte Werte .....	1047
5.7.46	chart.dataZoom.textStyle.fontFamily .....	1047
5.7.47	chart.dataZoom.textStyle.fontSize .....	1047
5.7.48	chart.dataZoom.textStyle.lineHeight .....	1047
5.7.49	chart.dataZoom.textStyle.width .....	1048
5.7.50	chart.dataZoom.textStyle.height .....	1049
5.7.51	chart.dataZoom.textStyle.textBorderColor .....	1049
5.7.52	chart.dataZoom.textStyle.textBorderWidth .....	1049
5.7.53	chart.dataZoom.textStyle.textShadowColor .....	1050
5.7.54	chart.dataZoom.textStyle.textShadowBlur .....	1050
5.7.55	chart.dataZoom.textStyle.textShadowOffsetX .....	1050
5.7.56	chart.dataZoom.textStyle.textShadowOffsetY .....	1051
5.7.57	chart.dataZoom.xAxisIndex .....	1051
5.7.58	chart.dataZoom.yAxisIndex .....	1052
5.7.59	chart.dataZoom.radiusAxisIndex .....	1053
5.7.60	chart.dataZoom.angleAxisIndex .....	1054
5.7.61	chart.dataZoom.filterMode .....	1055
	Erlaubte Werte .....	1057
5.7.62	chart.dataZoom.start .....	1057
5.7.63	chart.dataZoom.end .....	1058
5.7.64	chart.dataZoom.startValue .....	1058
5.7.65	chart.dataZoom.endValue .....	1058
5.7.66	chart.dataZoom.minSpan .....	1059
5.7.67	chart.dataZoom.maxSpan .....	1059
5.7.68	chart.dataZoom.minValueSpan .....	1060
5.7.69	chart.dataZoom.maxValueSpan .....	1060
5.7.70	chart.dataZoom.orient .....	1061
	Erlaubte Werte .....	1061
5.7.71	chart.dataZoom.zoomLock .....	1061
5.7.72	chart.dataZoom.throttle .....	1061
5.7.73	chart.dataZoom.rangeMode .....	1062
	Erlaubte Werte .....	1062

5.7.74	chart.dataZoom.zlevel.....	1062
5.7.75	chart.dataZoom.z.....	1063
5.7.76	chart.dataZoom.left.....	1063
	Erlaubte Werte .....	1064
5.7.77	chart.dataZoom.top .....	1064
	Erlaubte Werte .....	1064
5.7.78	chart.dataZoom.right .....	1064
	Erlaubte Werte .....	1065
5.7.79	chart.dataZoom.bottom.....	1065
	Erlaubte Werte .....	1065
5.7.80	chart.dataZoom.disabled .....	1066
5.7.81	chart.dataZoom.zoomOnMouseWheel.....	1066
	Erlaubte Werte .....	1066
5.7.82	chart.dataZoom.moveOnMouseMove .....	1067
	Erlaubte Werte .....	1067
5.7.83	chart.dataZoom.moveOnMouseWheel.....	1067
	Erlaubte Werte .....	1068
5.7.84	chart.dataZoom.preventDefaultMouseMove .....	1068
5.8	<b>ECharts (4) Gantt .....</b>	<b>1068</b>
5.8.1	chart.gantt.column.categoryname .....	1068
5.8.2	chart.gantt.column.categorygroup .....	1069
5.8.3	chart.gantt.column.barinfo .....	1069
5.8.4	chart.gantt.column.iconinfo .....	1070
5.8.5	chart.gantt.column.barcolor.....	1070
5.8.6	chart.gantt.screenitems .....	1071
5.8.7	chart.gantt.screenitems.height.....	1071
5.8.8	chart.gantt.iconlink.check.....	1072
5.8.9	chart.gantt.iconlink.path.....	1072
5.8.10	chart.gantt.icon.color .....	1073
5.8.11	chart.gantt.nodata.text .....	1073
5.8.12	chart.gantt.nodata.color .....	1074
5.8.13	chart.gantt.nodata.textColor .....	1074
5.8.14	chart.gantt.grid.height .....	1075
5.8.15	chart.gantt.grid.left.....	1075
5.8.16	chart.gantt.grid.top .....	1076
5.8.17	chart.gantt.grid.right .....	1076

5.8.18	chart.gantt.grid.bottom .....	1076
5.8.19	chart.gantt.nodata.maskColor.....	1077
5.8.20	chart.gantt.nodata.zlevel .....	1077
5.8.21	chart.gantt.nodata.fontSize .....	1078
5.8.22	chart.gantt.nodata.showSpinner.....	1078
5.8.23	chart.gantt.nodata.spinnerRadius.....	1079
5.8.24	chart.gantt.nodata.lineWidth.....	1079
5.8.25	chart.gantt.axis.path .....	1080
5.8.26	chart.gantt.axis.x .....	1080
5.8.27	chart.gantt.axis.y .....	1081
5.8.28	chart.gantt.axis.width.....	1081
5.8.29	chart.gantt.axis.height.....	1082
5.8.30	chart.gantt.axis.layout.....	1082
	Erlaubte Werte .....	1083
5.8.31	chart.gantt.axis.style.fill .....	1083
5.8.32	chart.gantt.axis.categorynametext.style.x.....	1083
5.8.33	chart.gantt.axis.categorynametext.style.y.....	1084
5.8.34	chart.gantt.axis.categorynametext.style.textVerticalAlign .....	1084
	Erlaubte Werte .....	1085
5.8.35	chart.gantt.axis.categorynametext.style.textAlign.....	1085
	Erlaubte Werte .....	1085
5.8.36	chart.gantt.axis.categorynametext.style.textFill .....	1085
5.8.37	chart.gantt.axis.categorygrouptext.style.x .....	1086
5.8.38	chart.gantt.axis.categorygrouptext.style.y .....	1086
5.8.39	chart.gantt.axis.categorygrouptext.style.textVerticalAlign.....	1087
	Erlaubte Werte .....	1087
5.8.40	chart.gantt.axis.categorygrouptext.style.textAlign .....	1088
	Erlaubte Werte .....	1088
5.8.41	chart.gantt.axis.categorygrouptext.style.textFill.....	1088
5.8.42	chart.gantt.rect.style.textFill.....	1089
5.8.43	chart.gantt.rect.style.stroke.....	1089
5.8.44	chart.gantt.rect.style.fill .....	1090
5.9	ECharts (4) General .....	1090
5.9.1	chart.type .....	1090
5.9.2	chart.theme .....	1091

5.10	ECharts (4) Geo.....	1091
5.10.1	chart.geo.object.....	1091
5.10.2	chart.geo.id.....	1092
5.10.3	chart.geo.show.....	1092
5.10.4	chart.geo.map.....	1093
5.10.5	chart.geo.roam.....	1094
	Erlaubte Werte.....	1094
5.10.6	chart.geo.center.....	1094
5.10.7	chart.geo.aspectScale.....	1095
5.10.8	chart.geo.boundingCoords.....	1095
5.10.9	chart.geo.zoom.....	1096
5.10.10	chart.geo.scaleLimit.object.....	1096
5.10.11	chart.geo.scaleLimit.min.....	1096
5.10.12	chart.geo.scaleLimit.max.....	1097
5.10.13	chart.geo.nameMap.....	1097
5.10.14	chart.geo.selectedMode.....	1098
5.10.15	chart.geo.label.object.....	1098
5.10.16	chart.geo.label.show.....	1098
5.10.17	chart.geo.label.position.....	1099
	Erlaubte Werte.....	1099
5.10.18	chart.geo.label.distance.....	1100
5.10.19	chart.geo.label.rotate.....	1100
5.10.20	chart.geo.label.offset.....	1100
5.10.21	chart.geo.label.formatter.....	1101
5.10.22	chart.geo.label.color.....	1102
5.10.23	chart.geo.label.fontStyle.....	1102
	Erlaubte Werte.....	1103
5.10.24	chart.geo.label.fontWeight.....	1103
	Erlaubte Werte.....	1103
5.10.25	chart.geo.label.fontFamily.....	1104
5.10.26	chart.geo.label.fontSize.....	1104
5.10.27	chart.geo.label.align.....	1104
	Erlaubte Werte.....	1105
5.10.28	chart.geo.label.verticalAlign.....	1105
	Erlaubte Werte.....	1106
5.10.29	chart.geo.label.lineHeight.....	1106

5.10.30	chart.geo.label.backgroundColor .....	1106
5.10.31	chart.geo.label.borderColor .....	1107
5.10.32	chart.geo.label.borderWidth .....	1107
5.10.33	chart.geo.label.borderRadius .....	1108
5.10.34	chart.geo.label.padding .....	1108
5.10.35	chart.geo.label.shadowColor .....	1108
5.10.36	chart.geo.label.shadowBlur .....	1109
5.10.37	chart.geo.label.shadowOffsetX .....	1109
5.10.38	chart.geo.label.shadowOffsetY .....	1110
5.10.39	chart.geo.label.width.....	1110
5.10.40	chart.geo.label.height.....	1110
5.10.41	chart.geo.label.textBorderColor .....	1111
5.10.42	chart.geo.label.textBorderWidth .....	1111
5.10.43	chart.geo.label.textShadowColor .....	1112
5.10.44	chart.geo.label.textShadowBlur .....	1112
5.10.45	chart.geo.label.textShadowOffsetX .....	1112
5.10.46	chart.geo.label.textShadowOffsetY .....	1113
5.10.47	chart.geo.label.rich.object .....	1113
5.10.48	chart.geo.label.emphasis.object.....	1114
5.10.49	chart.geo.label.emphasis.show .....	1115
5.10.50	chart.geo.label.emphasis.position .....	1115
	Erlaubte Werte .....	1115
5.10.51	chart.geo.label.emphasis.distance.....	1116
5.10.52	chart.geo.label.emphasis.rotate.....	1116
5.10.53	chart.geo.label.emphasis.offset.....	1116
5.10.54	chart.geo.label.emphasis.formatter.....	1117
5.10.55	chart.geo.label.emphasis.color.....	1118
5.10.56	chart.geo.label.emphasis.fontStyle.....	1118
	Erlaubte Werte .....	1119
5.10.57	chart.geo.label.emphasis.fontWeight.....	1119
	Erlaubte Werte .....	1119
5.10.58	chart.geo.label.emphasis.fontFamily .....	1120
5.10.59	chart.geo.label.emphasis.fontSize .....	1120
5.10.60	chart.geo.label.emphasis.align .....	1120
	Erlaubte Werte .....	1121
5.10.61	chart.geo.label.emphasis.verticalAlign .....	1121

	Erlaubte Werte .....	1122
5.10.62	chart.geo.label.emphasis.lineHeight.....	1122
5.10.63	chart.geo.label.emphasis.backgroundColor.....	1122
5.10.64	chart.geo.label.emphasis.borderColor.....	1123
5.10.65	chart.geo.label.emphasis.borderWidth.....	1123
5.10.66	chart.geo.label.emphasis.borderRadius .....	1124
5.10.67	chart.geo.label.emphasis.padding .....	1124
5.10.68	chart.geo.label.emphasis.shadowColor .....	1124
5.10.69	chart.geo.label.emphasis.shadowBlur .....	1125
5.10.70	chart.geo.label.emphasis.shadowOffsetX.....	1125
5.10.71	chart.geo.label.emphasis.shadowOffsetY .....	1126
5.10.72	chart.geo.label.emphasis.width .....	1126
5.10.73	chart.geo.label.emphasis.height .....	1126
5.10.74	chart.geo.label.emphasis.textBorderColor .....	1127
5.10.75	chart.geo.label.emphasis.textBorderWidth .....	1127
5.10.76	chart.geo.label.emphasis.textShadowColor .....	1128
5.10.77	chart.geo.label.emphasis.textShadowBlur .....	1128
5.10.78	chart.geo.label.emphasis.textShadowOffsetX.....	1128
5.10.79	chart.geo.label.emphasis.textShadowOffsetY .....	1129
5.10.80	chart.geo.label.emphasis.rich.object .....	1129
5.10.81	chart.geo.itemStyle.object.....	1130
5.10.82	chart.geo.itemStyle.areaColor.object .....	1131
5.10.83	chart.geo.itemStyle.areaColor.color .....	1131
5.10.84	chart.geo.itemStyle.areaColor.borderColor .....	1132
5.10.85	chart.geo.itemStyle.areaColor.borderWidth.....	1133
5.10.86	chart.geo.itemStyle.areaColor.borderType .....	1133
	Erlaubte Werte .....	1133
5.10.87	chart.geo.itemStyle.areaColor.shadowBlur.....	1134
5.10.88	chart.geo.itemStyle.areaColor.shadowColor.....	1134
5.10.89	chart.geo.itemStyle.areaColor.shadowOffsetX.....	1134
5.10.90	chart.geo.itemStyle.areaColor.shadowOffsetY .....	1135
5.10.91	chart.geo.itemStyle.areaColor.opacity .....	1135
5.10.92	chart.geo.itemStyle.emphasis.object.....	1135
5.10.93	chart.geo.itemStyle.emphasis.areaColor .....	1136
5.10.94	chart.geo.itemStyle.emphasis.color.....	1136
5.10.95	chart.geo.itemStyle.emphasis.borderColor.....	1137

5.10.96	chart.geo.itemStyle.emphasis.borderWidth .....	1138
5.10.97	chart.geo.itemStyle.emphasis.borderType.....	1138
	Erlaubte Werte .....	1138
5.10.98	chart.geo.itemStyle.emphasis.shadowBlur .....	1139
5.10.99	chart.geo.itemStyle.emphasis.shadowColor .....	1139
5.10.100	chart.geo.itemStyle.emphasis.shadowOffsetX .....	1139
5.10.101	chart.geo.itemStyle.emphasis.shadowOffsetY .....	1140
5.10.102	chart.geo.itemStyle.emphasis.opacity .....	1140
5.10.103	chart.geo.zlevel .....	1140
5.10.104	chart.geo.z.....	1141
5.10.105	chart.geo.left .....	1141
	Erlaubte Werte .....	1142
5.10.106	chart.geo.top.....	1142
	Erlaubte Werte .....	1142
5.10.107	chart.geo.right.....	1143
	Erlaubte Werte .....	1143
5.10.108	chart.geo.bottom .....	1143
	Erlaubte Werte .....	1144
5.10.109	chart.geo.layoutCenter.....	1144
5.10.110	chart.geo.layoutSize .....	1144
5.10.111	chart.geo.regions.{x}.object .....	1145
5.10.112	chart.geo.regions.{x}.name .....	1145
5.10.113	chart.geo.regions.{x}.selected.....	1145
5.10.114	chart.geo.regions.{x}.itemStyle.object .....	1146
5.10.115	chart.geo.regions.{x}.itemStyle.areaColor .....	1146
5.10.116	chart.geo.regions.{x}.itemStyle.color .....	1147
5.10.117	chart.geo.regions.{x}.itemStyle.borderColor .....	1148
5.10.118	chart.geo.regions.{x}.itemStyle.borderWidth .....	1149
5.10.119	chart.geo.regions.{x}.itemStyle.borderType .....	1149
	Erlaubte Werte .....	1149
5.10.120	chart.geo.regions.{x}.itemStyle.shadowBlur.....	1150
5.10.121	chart.geo.regions.{x}.itemStyle.shadowColor.....	1150
5.10.122	chart.geo.regions.{x}.itemStyle.shadowOffsetX .....	1150
5.10.123	chart.geo.regions.{x}.itemStyle.shadowOffsetY.....	1151
5.10.124	chart.geo.regions.{x}.itemStyle.opacity .....	1151
5.10.125	chart.geo.regions.{x}.itemStyle.emphasis.object .....	1151

5.10.126	chart.geo.regions.{x}.itemStyle.emphasis.areaColor .....	1152
5.10.127	chart.geo.regions.{x}.itemStyle.emphasis.color .....	1152
5.10.128	chart.geo.regions.{x}.itemStyle.emphasis.borderColor .....	1153
5.10.129	chart.geo.regions.{x}.itemStyle.emphasis.borderWidth .....	1154
5.10.130	chart.geo.regions.{x}.itemStyle.emphasis.borderType .....	1154
	Erlaubte Werte .....	1154
5.10.131	chart.geo.regions.{x}.itemStyle.emphasis.shadowBlur.....	1155
5.10.132	chart.geo.regions.{x}.itemStyle.emphasis.shadowColor .....	1155
5.10.133	chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetX .....	1155
5.10.134	chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetY.....	1156
5.10.135	chart.geo.regions.{x}.itemStyle.emphasis.opacity .....	1156
5.10.136	chart.geo.regions.{x}.label.object .....	1156
5.10.137	chart.geo.regions.{x}.label.show.....	1157
5.10.138	chart.geo.regions.{x}.label.position.....	1157
	Erlaubte Werte .....	1158
5.10.139	chart.geo.regions.{x}.label.distance .....	1158
5.10.140	chart.geo.regions.{x}.label.rotate .....	1158
5.10.141	chart.geo.regions.{x}.label.offset .....	1159
5.10.142	chart.geo.regions.{x}.label.formatter .....	1159
5.10.143	chart.geo.regions.{x}.label.color .....	1160
5.10.144	chart.geo.regions.{x}.label.fontStyle .....	1161
	Erlaubte Werte .....	1161
5.10.145	chart.geo.regions.{x}.label.fontWeight .....	1161
	Erlaubte Werte .....	1162
5.10.146	chart.geo.regions.{x}.label.fontFamily .....	1162
5.10.147	chart.geo.regions.{x}.label.fontSize.....	1162
5.10.148	chart.geo.regions.{x}.label.align .....	1163
	Erlaubte Werte .....	1163
5.10.149	chart.geo.regions.{x}.label.verticalAlign.....	1163
	Erlaubte Werte .....	1164
5.10.150	chart.geo.regions.{x}.label.lineHeight .....	1164
5.10.151	chart.geo.regions.{x}.label.backgroundColor .....	1164
5.10.152	chart.geo.regions.{x}.label.borderColor .....	1165
5.10.153	chart.geo.regions.{x}.label.borderWidth .....	1166
5.10.154	chart.geo.regions.{x}.label.borderRadius.....	1166
5.10.155	chart.geo.regions.{x}.label.padding.....	1166



5.10.156	chart.geo.regions.{x}.label.shadowColor .....	1167
5.10.157	chart.geo.regions.{x}.label.shadowBlur .....	1167
5.10.158	chart.geo.regions.{x}.label.shadowOffsetX .....	1167
5.10.159	chart.geo.regions.{x}.label.shadowOffsetY .....	1168
5.10.160	chart.geo.regions.{x}.label.width .....	1168
5.10.161	chart.geo.regions.{x}.label.height .....	1169
5.10.162	chart.geo.regions.{x}.label.textBorderColor .....	1169
5.10.163	chart.geo.regions.{x}.label.textBorderWidth .....	1169
5.10.164	chart.geo.regions.{x}.label.textShadowColor .....	1170
5.10.165	chart.geo.regions.{x}.label.textShadowBlur .....	1170
5.10.166	chart.geo.regions.{x}.label.textShadowOffsetX .....	1171
5.10.167	chart.geo.regions.{x}.label.textShadowOffsetY .....	1171
5.10.168	chart.geo.regions.{x}.label.rich.object .....	1171
5.10.169	chart.geo.regions.{x}.label.emphasis.object .....	1172
5.10.170	chart.geo.regions.{x}.label.emphasis.show .....	1173
5.10.171	chart.geo.regions.{x}.label.emphasis.position .....	1173
	Erlaubte Werte .....	1174
5.10.172	chart.geo.regions.{x}.label.emphasis.distance .....	1174
5.10.173	chart.geo.regions.{x}.label.emphasis.rotate .....	1174
5.10.174	chart.geo.regions.{x}.label.emphasis.offset .....	1175
5.10.175	chart.geo.regions.{x}.label.emphasis.formatter .....	1175
5.10.176	chart.geo.regions.{x}.label.emphasis.color .....	1176
5.10.177	chart.geo.regions.{x}.label.emphasis.fontStyle .....	1176
	Erlaubte Werte .....	1177
5.10.178	chart.geo.regions.{x}.label.emphasis.fontWeight .....	1177
	Erlaubte Werte .....	1177
5.10.179	chart.geo.regions.{x}.label.emphasis.fontFamily .....	1178
5.10.180	chart.geo.regions.{x}.label.emphasis.fontSize .....	1178
5.10.181	chart.geo.regions.{x}.label.emphasis.align .....	1178
	Erlaubte Werte .....	1179
5.10.182	chart.geo.regions.{x}.label.emphasis.verticalAlign .....	1179
	Erlaubte Werte .....	1180
5.10.183	chart.geo.regions.{x}.label.emphasis.lineHeight .....	1180
5.10.184	chart.geo.regions.{x}.label.emphasis.backgroundColor .....	1180
5.10.185	chart.geo.regions.{x}.label.emphasis.borderColor .....	1181
5.10.186	chart.geo.regions.{x}.label.emphasis.borderWidth .....	1181

5.10.187	chart.geo.regions.{x}.label.emphasis.borderRadius.....	1182
5.10.188	chart.geo.regions.{x}.label.emphasis.padding.....	1182
5.10.189	chart.geo.regions.{x}.label.emphasis.shadowColor .....	1182
5.10.190	chart.geo.regions.{x}.label.emphasis.shadowBlur .....	1183
5.10.191	chart.geo.regions.{x}.label.emphasis.shadowOffsetX .....	1183
5.10.192	chart.geo.regions.{x}.label.emphasis.shadowOffsetY .....	1184
5.10.193	chart.geo.regions.{x}.label.emphasis.width.....	1184
5.10.194	chart.geo.regions.{x}.label.emphasis.height.....	1184
5.10.195	chart.geo.regions.{x}.label.emphasis.textBorderColor .....	1185
5.10.196	chart.geo.regions.{x}.label.emphasis.textBorderWidth .....	1185
5.10.197	chart.geo.regions.{x}.label.emphasis.textShadowColor .....	1186
5.10.198	chart.geo.regions.{x}.label.emphasis.textShadowBlur .....	1186
5.10.199	chart.geo.regions.{x}.label.emphasis.textShadowOffsetX .....	1186
5.10.200	chart.geo.regions.{x}.label.emphasis.textShadowOffsetY .....	1187
5.10.201	chart.geo.regions.{x}.label.emphasis.rich.object .....	1187
5.10.202	chart.geo.silent .....	1188
5.11	ECharts (4) Graphic .....	1189
5.11.1	chart.graphic.object .....	1189
5.12	ECharts (4) Grid .....	1196
5.12.1	chart.grid.object.....	1196
5.12.2	chart.grid.{x}.id.....	1196
5.12.3	chart.grid.{x}.show .....	1197
5.12.4	chart.grid.{x}.zlevel .....	1197
5.12.5	chart.grid.{x}.z .....	1197
5.12.6	chart.grid.{x}.left .....	1198
	Erlaubte Werte .....	1198
5.12.7	chart.grid.{x}.top .....	1198
	Erlaubte Werte .....	1199
5.12.8	chart.grid.{x}.right.....	1199
	Erlaubte Werte .....	1199
5.12.9	chart.grid.{x}.bottom .....	1200
	Erlaubte Werte .....	1200
5.12.10	chart.grid.{x}.width .....	1200
5.12.11	chart.grid.{x}.height .....	1200
5.12.12	chart.grid.{x}.containLabel .....	1201

5.12.13	chart.grid.{x}.backgroundColor.....	1201
5.12.14	chart.grid.{x}.borderColor .....	1202
5.12.15	chart.grid.{x}.borderWidth.....	1202
5.12.16	chart.grid.{x}.shadowBlur.....	1203
5.12.17	chart.grid.{x}.shadowColor.....	1203
5.12.18	chart.grid.{x}.shadowOffsetX.....	1203
5.12.19	chart.grid.{x}.shadowOffsetY.....	1204
5.12.20	chart.grid.{x}.tooltip.object .....	1204
5.12.21	chart.grid.{x}.tooltip.show.....	1205
5.12.22	chart.grid.{x}.tooltip.trigger .....	1205
	Erlaubte Werte .....	1206
5.12.23	chart.grid.{x}.tooltip.axisPointer.object .....	1206
5.12.24	chart.grid.{x}.tooltip.axisPointer.type .....	1206
	Erlaubte Werte .....	1207
5.12.25	chart.grid.{x}.tooltip.axisPointer.axis .....	1207
	Erlaubte Werte .....	1207
5.12.26	chart.grid.{x}.tooltip.axisPointer.snap.....	1207
5.12.27	chart.grid.{x}.tooltip.axisPointer.z.....	1208
5.12.28	chart.grid.{x}.tooltip.axisPointer.label.object.....	1208
5.12.29	chart.grid.{x}.tooltip.axisPointer.label.show .....	1208
5.12.30	chart.grid.{x}.tooltip.axisPointer.label.precision .....	1209
5.12.31	chart.grid.{x}.tooltip.axisPointer.label.formatter .....	1209
5.12.32	chart.grid.{x}.tooltip.axisPointer.label.margin .....	1211
5.12.33	chart.grid.{x}.tooltip.axisPointer.label.color.....	1211
5.12.34	chart.grid.{x}.tooltip.axisPointer.label.fontStyle .....	1211
5.12.35	chart.grid.{x}.tooltip.axisPointer.label.fontWeight.....	1212
	Erlaubte Werte .....	1212
5.12.36	chart.grid.{x}.tooltip.axisPointer.label.fontFamily .....	1212
5.12.37	chart.grid.{x}.tooltip.axisPointer.label.fontSize.....	1213
5.12.38	chart.grid.{x}.tooltip.axisPointer.label.lineHeight .....	1213
5.12.39	chart.grid.{x}.tooltip.axisPointer.label.width.....	1214
5.12.40	chart.grid.{x}.tooltip.axisPointer.label.height.....	1214
5.12.41	chart.grid.{x}.tooltip.axisPointer.label.textBorderColor .....	1214
5.12.42	chart.grid.{x}.tooltip.axisPointer.label.textBorderWidth .....	1215
5.12.43	chart.grid.{x}.tooltip.axisPointer.label.textShadowColor .....	1215
5.12.44	chart.grid.{x}.tooltip.axisPointer.label.textShadowBlur .....	1216

5.12.45	chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetX .....	1216
5.12.46	chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetY .....	1216
5.12.47	chart.grid.{x}.tooltip.axisPointer.label.padding .....	1217
5.12.48	chart.grid.{x}.tooltip.axisPointer.label.backgroundColor .....	1217
5.12.49	chart.grid.{x}.tooltip.axisPointer.label.borderColor .....	1218
5.12.50	chart.grid.{x}.tooltip.axisPointer.label.borderWidth .....	1218
5.12.51	chart.grid.{x}.tooltip.axisPointer.label.shadowBlur .....	1218
5.12.52	chart.grid.{x}.tooltip.axisPointer.label.shadowColor .....	1219
5.12.53	chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetX .....	1219
5.12.54	chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetY .....	1220
5.12.55	chart.grid.{x}.tooltip.axisPointer.lineStyle.object.....	1220
5.12.56	chart.grid.{x}.tooltip.axisPointer.lineStyle.color .....	1220
5.12.57	chart.grid.{x}.tooltip.axisPointer.lineStyle.width .....	1222
5.12.58	chart.grid.{x}.tooltip.axisPointer.lineStyle.type.....	1222
	Erlaubte Werte .....	1222
5.12.59	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowBlur .....	1222
5.12.60	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowColor .....	1223
5.12.61	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetX.....	1223
5.12.62	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetY .....	1224
5.12.63	chart.grid.{x}.tooltip.axisPointer.lineStyle.opacity.....	1224
5.12.64	chart.grid.{x}.tooltip.axisPointer.shadowStyle.object.....	1224
5.12.65	chart.grid.{x}.tooltip.axisPointer.shadowStyle.color .....	1225
5.12.66	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowBlur .....	1226
5.12.67	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowColor .....	1227
5.12.68	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetX.....	1227
5.12.69	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	1228
5.12.70	chart.grid.{x}.tooltip.axisPointer.shadowStyle.opacity.....	1228
5.12.71	chart.grid.{x}.tooltip.axisPointer.crossStyle.object.....	1228
5.12.72	chart.grid.{x}.tooltip.axisPointer.crossStyle.color .....	1229
5.12.73	chart.grid.{x}.tooltip.axisPointer.crossStyle.width .....	1230
5.12.74	chart.grid.{x}.tooltip.axisPointer.crossStyle.type .....	1231
	Erlaubte Werte .....	1231
5.12.75	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowBlur .....	1231
5.12.76	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowColor .....	1232
5.12.77	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetX .....	1232
5.12.78	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetY .....	1232

5.12.79	chart.grid.{x}.tooltip.axisPointer.crossStyle.opacity .....	1233
5.12.80	chart.grid.{x}.tooltip.position .....	1233
5.12.81	chart.grid.{x}.tooltip.formatter .....	1235
5.12.82	chart.grid.{x}.tooltip.backgroundColor .....	1237
5.12.83	chart.grid.{x}.tooltip.borderColor .....	1237
5.12.84	chart.grid.{x}.tooltip.borderWidth .....	1238
5.12.85	chart.grid.{x}.tooltip.padding.....	1238
5.12.86	chart.grid.{x}.tooltip.textStyle.object .....	1239
5.12.87	chart.grid.{x}.tooltip.textStyle.color .....	1239
5.12.88	chart.grid.{x}.tooltip.textStyle.fontStyle .....	1239
	Erlaubte Werte .....	1240
5.12.89	chart.grid.{x}.tooltip.textStyle.fontWeight .....	1240
	Erlaubte Werte .....	1240
5.12.90	chart.grid.{x}.tooltip.textStyle.fontFamily.....	1240
5.12.91	chart.grid.{x}.tooltip.textStyle.fontSize .....	1241
5.12.92	chart.grid.{x}.tooltip.textStyle.lineHeight .....	1241
5.12.93	chart.grid.{x}.tooltip.textStyle.width .....	1242
5.12.94	chart.grid.{x}.tooltip.textStyle.height .....	1242
5.12.95	chart.grid.{x}.tooltip.textStyle.textBorderColor.....	1243
5.12.96	chart.grid.{x}.tooltip.textStyle.textBorderWidth.....	1243
5.12.97	chart.grid.{x}.tooltip.textStyle.textShadowColor .....	1243
5.12.98	chart.grid.{x}.tooltip.textStyle.textShadowBlur.....	1244
5.12.99	chart.grid.{x}.tooltip.textStyle.textShadowOffsetX .....	1244
5.12.100	chart.grid.{x}.tooltip.textStyle.textShadowOffsetY.....	1244
5.12.101	chart.grid.{x}.tooltip.extraCssText .....	1245
5.13	<b>ECharts (4) Legend .....</b>	<b>1245</b>
5.13.1	chart.legend.object.....	1245
5.13.2	chart.legend.type.....	1246
	Erlaubte Werte .....	1246
5.13.3	chart.legend.id .....	1247
5.13.4	chart.legend.show .....	1247
5.13.5	chart.legend.zlevel.....	1247
5.13.6	chart.legend.z.....	1248
5.13.7	chart.legend.left.....	1248
	Erlaubte Werte .....	1249
5.13.8	chart.legend.top .....	1249

	Erlaubte Werte .....	1249
5.13.9	chart.legend.right .....	1249
5.13.10	chart.legend.bottom.....	1250
	Erlaubte Werte .....	1250
5.13.11	chart.legend.width.....	1250
5.13.12	chart.legend.height.....	1251
5.13.13	chart.legend.orient .....	1251
	Erlaubte Werte .....	1251
5.13.14	chart.legend.align .....	1251
	Erlaubte Werte .....	1252
5.13.15	chart.legend.padding .....	1252
5.13.16	chart.legend.itemGap .....	1253
5.13.17	chart.legend.itemWidth.....	1253
5.13.18	chart.legend.itemHeight .....	1254
5.13.19	chart.legend.symbolKeepAspect .....	1254
5.13.20	chart.legend.formatter .....	1254
5.13.21	chart.legend.selectedMode.....	1255
5.13.22	chart.legend.inactiveColor .....	1255
5.13.23	chart.legend.selected .....	1256
5.13.24	chart.legend.textStyle.object.....	1256
5.13.25	chart.legend.textStyle.color.....	1256
5.13.26	chart.legend.textStyle.fontStyle .....	1257
	Erlaubte Werte .....	1257
5.13.27	chart.legend.textStyle.fontWeight.....	1257
5.13.28	chart.legend.textStyle.fontFamily .....	1258
5.13.29	chart.legend.textStyle.fontSize.....	1258
5.13.30	chart.legend.textStyle.align .....	1259
	Erlaubte Werte .....	1259
5.13.31	chart.legend.textStyle.verticalAlign .....	1259
	Erlaubte Werte .....	1260
5.13.32	chart.legend.textStyle.lineHeight .....	1260
5.13.33	chart.legend.textStyle.backgroundColor .....	1261
5.13.34	chart.legend.textStyle.borderColor .....	1261
5.13.35	chart.legend.textStyle.borderWidth .....	1262
5.13.36	chart.legend.textStyle.borderRadius.....	1262
5.13.37	chart.legend.textStyle.padding .....	1262

5.13.38	chart.legend.textStyle.shadowColor .....	1263
5.13.39	chart.legend.textStyle.shadowBlur .....	1263
5.13.40	chart.legend.textStyle.shadowOffsetX .....	1263
5.13.41	chart.legend.textStyle.shadowOffsetY .....	1264
5.13.42	chart.legend.textStyle.width.....	1264
	Erlaubte Werte .....	1265
5.13.43	chart.legend.textStyle.height.....	1265
5.13.44	chart.legend.textStyle.textBorderColor .....	1265
5.13.45	chart.legend.textStyle.textBorderWidth .....	1266
5.13.46	chart.legend.textStyle.textShadowColor .....	1266
5.13.47	chart.legend.textStyle.textShadowBlur .....	1266
5.13.48	chart.legend.textStyle.textShadowOffsetX .....	1267
5.13.49	chart.legend.textStyle.textShadowOffsetY .....	1267
5.13.50	chart.legend.textStyle.rich .....	1267
5.13.51	chart.legend.tooltip .....	1268
5.13.52	chart.legend.data.object .....	1269
5.13.53	chart.legend.backgroundColor .....	1269
5.13.54	chart.legend.borderColor.....	1270
5.13.55	chart.legend.borderWidth.....	1270
5.13.56	chart.legend.borderRadius .....	1271
5.13.57	chart.legend.shadowBlur .....	1271
5.13.58	chart.legend.shadowColor .....	1271
5.13.59	chart.legend.shadowOffsetX.....	1272
5.13.60	chart.legend.shadowOffsetY .....	1272
5.13.61	chart.legend.scrollDataIndex .....	1273
5.13.62	chart.legend.pageButtonItemGap.....	1273
5.13.63	chart.legend.pageButtonGap.....	1273
5.13.64	chart.legend.pageButtonPosition.....	1274
5.13.65	chart.legend.pageFormatter .....	1274
5.13.66	chart.legend.pageIcons .....	1275
5.13.67	chart.legend.pageIcons.horizontal.....	1275
5.13.68	chart.legend.pageIcons.vertical.....	1276
5.13.69	chart.legend.pageIconColor.....	1276
5.13.70	chart.legend.pageIconInactiveColor .....	1277
5.13.71	chart.legend.pageIconSize.....	1277
5.13.72	chart.legend.pageTextStyle.object.....	1277

5.13.73	chart.legend.pageTextStyle.color.....	1278
5.13.74	chart.legend.pageTextStyle.fontStyle.....	1278
	Erlaubte Werte .....	1279
5.13.75	chart.legend.pageTextStyle.fontWeight.....	1279
5.13.76	chart.legend.pageTextStyle.fontFamily .....	1279
5.13.77	chart.legend.pageTextStyle.fontSize.....	1280
5.13.78	chart.legend.pageTextStyle.lineHeight .....	1280
5.13.79	chart.legend.pageTextStyle.width.....	1280
	Erlaubte Werte .....	1281
5.13.80	chart.legend.pageTextStyle.height.....	1281
5.13.81	chart.legend.pageTextStyle.textBorderColor .....	1281
5.13.82	chart.legend.pageTextStyle.textBorderWidth .....	1282
5.13.83	chart.legend.pageTextStyle.textShadowColor .....	1282
5.13.84	chart.legend.pageTextStyle.textShadowBlur .....	1283
5.13.85	chart.legend.pageTextStyle.textShadowOffsetX .....	1283
5.13.86	chart.legend.pageTextStyle.textShadowOffsetY .....	1283
5.13.87	chart.legend.animation .....	1284
5.13.88	chart.legend.animationDurationUpdate.....	1284
5.14	ECharts (4) Parallel.....	1284
5.14.1	chart.parallel.object .....	1284
5.14.2	chart.parallel.id.....	1287
5.14.3	chart.parallel.zlevel .....	1288
5.14.4	chart.parallel.z .....	1288
5.14.5	chart.parallel.left .....	1289
	Erlaubte Werte .....	1289
5.14.6	chart.parallel.top .....	1289
	Erlaubte Werte .....	1290
5.14.7	chart.parallel.right .....	1290
	Erlaubte Werte .....	1290
5.14.8	chart.parallel.bottom .....	1290
5.14.9	chart.parallel.width .....	1291
5.14.10	chart.parallel.height .....	1291
5.14.11	chart.parallel.layout .....	1291
	Erlaubte Werte .....	1292
5.14.12	chart.parallel.axisExpandable.....	1292
5.14.13	chart.parallel.axisExpandCenter.....	1292



5.14.14	chart.parallel.axisExpandCount.....	1293
5.14.15	chart.parallel.axisExpandWidth.....	1293
5.14.16	chart.parallel.axisExpandTriggerOn.....	1294
	Erlaubte Werte.....	1294
5.14.17	chart.parallel.parallelAxisDefault.object.....	1294
5.14.18	chart.parallel.parallelAxisDefault.type.....	1294
	Erlaubte Werte.....	1295
5.14.19	chart.parallel.parallelAxisDefault.name.....	1295
5.14.20	chart.parallel.parallelAxisDefault.nameLocation.....	1295
	Erlaubte Werte.....	1296
5.14.21	chart.parallel.parallelAxisDefault.nameTextStyle.object.....	1296
5.14.22	chart.parallel.parallelAxisDefault.nameTextStyle.color.....	1296
5.14.23	chart.parallel.parallelAxisDefault.nameTextStyle.fontStyle.....	1297
	Erlaubte Werte.....	1297
5.14.24	chart.parallel.parallelAxisDefault.nameTextStyle.fontWeight.....	1297
	Erlaubte Werte.....	1297
5.14.25	chart.parallel.parallelAxisDefault.nameTextStyle.fontFamily.....	1298
5.14.26	chart.parallel.parallelAxisDefault.nameTextStyle.fontSize.....	1298
5.14.27	chart.parallel.parallelAxisDefault.nameTextStyle.align.....	1298
	Erlaubte Werte.....	1299
5.14.28	chart.parallel.parallelAxisDefault.nameTextStyle.verticalAlign.....	1299
	Erlaubte Werte.....	1300
5.14.29	chart.parallel.parallelAxisDefault.nameTextStyle.lineHeight.....	1300
5.14.30	chart.parallel.parallelAxisDefault.nameTextStyle.backgroundColor.....	1300
5.14.31	chart.parallel.parallelAxisDefault.nameTextStyle.borderColor.....	1301
5.14.32	chart.parallel.parallelAxisDefault.nameTextStyle.borderWidth.....	1301
5.14.33	chart.parallel.parallelAxisDefault.nameTextStyle.borderRadius.....	1302
5.14.34	chart.parallel.parallelAxisDefault.nameTextStyle.padding.....	1302
5.14.35	chart.parallel.parallelAxisDefault.nameTextStyle.shadowColor.....	1303
5.14.36	chart.parallel.parallelAxisDefault.nameTextStyle.shadowBlur.....	1303
5.14.37	chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetX.....	1303
5.14.38	chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetY.....	1304
5.14.39	chart.parallel.parallelAxisDefault.nameTextStyle.width.....	1304
5.14.40	chart.parallel.parallelAxisDefault.nameTextStyle.height.....	1305
5.14.41	chart.parallel.parallelAxisDefault.nameTextStyle.textBorderColor.....	1305
5.14.42	chart.parallel.parallelAxisDefault.nameTextStyle.textBorderWidth.....	1305

5.14.43	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowColor .....	1306
5.14.44	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowBlur .....	1306
5.14.45	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetX .....	1306
5.14.46	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetY .....	1307
5.14.47	chart.parallel.parallelAxisDefault.nameTextStyle.rich.object .....	1307
5.14.48	chart.parallel.parallelAxisDefault.nameGap .....	1308
5.14.49	chart.parallel.parallelAxisDefault.nameRotate .....	1309
5.14.50	chart.parallel.parallelAxisDefault.inverse .....	1309
5.14.51	chart.parallel.parallelAxisDefault.boundaryGap .....	1309
5.14.52	chart.parallel.parallelAxisDefault.min .....	1310
5.14.53	chart.parallel.parallelAxisDefault.max .....	1310
5.14.54	chart.parallel.parallelAxisDefault.scale .....	1311
5.14.55	chart.parallel.parallelAxisDefault.splitNumber .....	1311
5.14.56	chart.parallel.parallelAxisDefault.minInterval .....	1312
5.14.57	chart.parallel.parallelAxisDefault.interval .....	1312
5.14.58	chart.parallel.parallelAxisDefault.logBase .....	1313
5.14.59	chart.parallel.parallelAxisDefault.silent .....	1313
5.14.60	chart.parallel.parallelAxisDefault.triggerEvent .....	1313
5.14.61	chart.parallel.parallelAxisDefault.axisLine.object .....	1314
5.14.62	chart.parallel.parallelAxisDefault.axisLine.show .....	1314
5.14.63	chart.parallel.parallelAxisDefault.axisLine.symbol .....	1315
	Erlaubte Werte .....	1315
5.14.64	chart.parallel.parallelAxisDefault.axisLine.symbolSize .....	1315
5.14.65	chart.parallel.parallelAxisDefault.axisLine.symbolOffset .....	1316
5.14.66	chart.parallel.parallelAxisDefault.axisLine.lineStyle.object .....	1316
5.14.67	chart.parallel.parallelAxisDefault.axisLine.lineStyle.color .....	1316
5.14.68	chart.parallel.parallelAxisDefault.axisLine.lineStyle.width .....	1317
5.14.69	chart.parallel.parallelAxisDefault.axisLine.lineStyle.type .....	1318
	Erlaubte Werte .....	1318
5.14.70	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowBlur .....	1318
5.14.71	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowColor .....	1319
5.14.72	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetX .....	1319
5.14.73	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetY .....	1319
5.14.74	chart.parallel.parallelAxisDefault.axisLine.lineStyle.opacity .....	1320
5.14.75	chart.parallel.parallelAxisDefault.axisTick.object .....	1320
5.14.76	chart.parallel.parallelAxisDefault.axisTick.show .....	1321

5.14.77	chart.parallel.parallelAxisDefault.axisTick.alignWithLabel .....	1321
5.14.78	chart.parallel.parallelAxisDefault.axisTick.interval .....	1321
5.14.79	chart.parallel.parallelAxisDefault.axisTick.inside .....	1322
5.14.80	chart.parallel.parallelAxisDefault.axisTick.length .....	1322
5.14.81	chart.parallel.parallelAxisDefault.axisTick.lineStyle.object .....	1322
5.14.82	chart.parallel.parallelAxisDefault.axisTick.lineStyle.color .....	1323
5.14.83	chart.parallel.parallelAxisDefault.axisTick.lineStyle.width .....	1323
5.14.84	chart.parallel.parallelAxisDefault.axisTick.lineStyle.type .....	1323
	Erlaubte Werte .....	1324
5.14.85	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowBlur .....	1324
5.14.86	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowColor .....	1324
5.14.87	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetX .....	1325
5.14.88	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetY .....	1325
5.14.89	chart.parallel.parallelAxisDefault.axisTick.lineStyle.opacity .....	1325
5.14.90	chart.parallel.parallelAxisDefault.axisLabel.object .....	1326
5.14.91	chart.parallel.parallelAxisDefault.axisLabel.show .....	1326
5.14.92	chart.parallel.parallelAxisDefault.axisLabel.interval .....	1326
5.14.93	chart.parallel.parallelAxisDefault.axisLabel.inside .....	1327
5.14.94	chart.parallel.parallelAxisDefault.axisLabel.rotate .....	1327
5.14.95	chart.parallel.parallelAxisDefault.axisLabel.margin .....	1328
5.14.96	chart.parallel.parallelAxisDefault.axisLabel.formatter .....	1328
5.14.97	chart.parallel.parallelAxisDefault.axisLabel.showMinLabel .....	1329
5.14.98	chart.parallel.parallelAxisDefault.axisLabel.showMaxLabel .....	1329
5.14.99	chart.parallel.parallelAxisDefault.axisLabel.color .....	1330
5.14.100	chart.parallel.parallelAxisDefault.axisLabel.fontStyle .....	1330
	Erlaubte Werte .....	1331
5.14.101	chart.parallel.parallelAxisDefault.axisLabel.fontWeight .....	1331
	Erlaubte Werte .....	1331
5.14.102	chart.parallel.parallelAxisDefault.axisLabel.fontFamily .....	1331
5.14.103	chart.parallel.parallelAxisDefault.axisLabel.fontSize .....	1332
5.14.104	chart.parallel.parallelAxisDefault.axisLabel.align .....	1332
	Erlaubte Werte .....	1333
5.14.105	chart.parallel.parallelAxisDefault.axisLabel.verticalAlign .....	1333
	Erlaubte Werte .....	1333
5.14.106	chart.parallel.parallelAxisDefault.axisLabel.lineHeight .....	1334
5.14.107	chart.parallel.parallelAxisDefault.axisLabel.backgroundColor .....	1334

5.14.108	chart.parallel.parallelAxisDefault.axisLabel.borderColor .....	1335
5.14.109	chart.parallel.parallelAxisDefault.axisLabel.borderWidth .....	1335
5.14.110	chart.parallel.parallelAxisDefault.axisLabel.borderRadius .....	1336
5.14.111	chart.parallel.parallelAxisDefault.axisLabel.padding .....	1336
5.14.112	chart.parallel.parallelAxisDefault.axisLabel.shadowColor .....	1336
5.14.113	chart.parallel.parallelAxisDefault.axisLabel.shadowBlur .....	1337
5.14.114	chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetX .....	1337
5.14.115	chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetY .....	1337
5.14.116	chart.parallel.parallelAxisDefault.axisLabel.width .....	1338
5.14.117	chart.parallel.parallelAxisDefault.axisLabel.height .....	1338
5.14.118	chart.parallel.parallelAxisDefault.axisLabel.textBorderColor .....	1339
5.14.119	chart.parallel.parallelAxisDefault.axisLabel.textBorderWidth .....	1339
5.14.120	chart.parallel.parallelAxisDefault.axisLabel.textShadowColor .....	1339
5.14.121	chart.parallel.parallelAxisDefault.axisLabel.textShadowBlur .....	1340
5.14.122	chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetX .....	1340
5.14.123	chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetY .....	1341
5.14.124	chart.parallel.parallelAxisDefault.axisLabel.rich.object .....	1341
5.15	<b>ECharts (4) Parallel Axis .....</b>	<b>1342</b>
5.15.1	chart.parallelAxis.object .....	1342
5.15.2	chart.parallelAxis.id .....	1345
5.15.3	chart.parallelAxis.dim .....	1346
5.15.4	chart.parallelAxis.parallelIndex .....	1346
5.15.5	chart.parallelAxis.realtime .....	1347
5.15.6	chart.parallelAxis.areaSelectStyle.object .....	1348
5.15.7	chart.parallelAxis.areaSelectStyle.width .....	1348
5.15.8	chart.parallelAxis.areaSelectStyle.borderWidth .....	1348
5.15.9	chart.parallelAxis.areaSelectStyle.borderColor .....	1349
5.15.10	chart.parallelAxis.areaSelectStyle.color .....	1349
5.15.11	chart.parallelAxis.areaSelectStyle.opacity .....	1349
5.15.12	chart.parallelAxis.type .....	1350
	Erlaubte Werte .....	1350
5.15.13	chart.parallelAxis.name .....	1350
5.15.14	chart.parallelAxis.nameLocation .....	1351
	Erlaubte Werte .....	1351
5.15.15	chart.parallelAxis.nameTextStyle.object .....	1351
5.15.16	chart.parallelAxis.nameTextStyle.color .....	1352

5.15.17	chart.parallelAxis.nameTextStyle.fontStyle .....	1352
	Erlaubte Werte .....	1352
5.15.18	chart.parallelAxis.nameTextStyle.fontWeight .....	1352
	Erlaubte Werte .....	1353
5.15.19	chart.parallelAxis.nameTextStyle.fontFamily .....	1353
5.15.20	chart.parallelAxis.nameTextStyle.fontSize .....	1353
5.15.21	chart.parallelAxis.nameTextStyle.align .....	1354
	Erlaubte Werte .....	1354
5.15.22	chart.parallelAxis.nameTextStyle.verticalAlign .....	1355
	Erlaubte Werte .....	1355
5.15.23	chart.parallelAxis.nameTextStyle.lineHeight .....	1355
5.15.24	chart.parallelAxis.nameTextStyle.backgroundColor .....	1356
5.15.25	chart.parallelAxis.nameTextStyle.borderColor .....	1356
5.15.26	chart.parallelAxis.nameTextStyle.borderWidth .....	1357
5.15.27	chart.parallelAxis.nameTextStyle.borderRadius .....	1357
5.15.28	chart.parallelAxis.nameTextStyle.padding .....	1358
5.15.29	chart.parallelAxis.nameTextStyle.shadowColor .....	1358
5.15.30	chart.parallelAxis.nameTextStyle.shadowBlur .....	1358
5.15.31	chart.parallelAxis.nameTextStyle.shadowOffsetX .....	1359
5.15.32	chart.parallelAxis.nameTextStyle.shadowOffsetY .....	1359
5.15.33	chart.parallelAxis.nameTextStyle.width .....	1359
5.15.34	chart.parallelAxis.nameTextStyle.height .....	1360
5.15.35	chart.parallelAxis.nameTextStyle.textBorderColor .....	1360
5.15.36	chart.parallelAxis.nameTextStyle.textBorderWidth .....	1361
5.15.37	chart.parallelAxis.nameTextStyle.textShadowColor .....	1361
5.15.38	chart.parallelAxis.nameTextStyle.textShadowBlur .....	1361
5.15.39	chart.parallelAxis.nameTextStyle.textShadowOffsetX .....	1362
5.15.40	chart.parallelAxis.nameTextStyle.textShadowOffsetY .....	1362
5.15.41	chart.parallelAxis.nameTextStyle.rich.object .....	1363
5.15.42	chart.parallelAxis.nameGap .....	1364
5.15.43	chart.parallelAxis.nameRotate .....	1364
5.15.44	chart.parallelAxis.inverse .....	1364
5.15.45	chart.parallelAxis.boundaryGap .....	1365
5.15.46	chart.parallelAxis.min .....	1365
5.15.47	chart.parallelAxis.max .....	1366
5.15.48	chart.parallelAxis.scale .....	1366

5.15.49	chart.parallelAxis.splitNumber .....	1367
5.15.50	chart.parallelAxis.minInterval .....	1367
5.15.51	chart.parallelAxis.interval .....	1367
5.15.52	chart.parallelAxis.logBase .....	1368
5.15.53	chart.parallelAxis.silent .....	1368
5.15.54	chart.parallelAxis.triggerEvent.....	1369
5.15.55	chart.parallelAxis.axisLine.object .....	1369
5.15.56	chart.parallelAxis.axisLine.show .....	1370
5.15.57	chart.parallelAxis.axisLine.symbol .....	1370
	Erlaubte Werte .....	1370
5.15.58	chart.parallelAxis.axisLine.symbolSize.....	1371
5.15.59	chart.parallelAxis.axisLine.symbolOffset .....	1371
5.15.60	chart.parallelAxis.axisLine.lineStyle.object.....	1371
5.15.61	chart.parallelAxis.axisLine.lineStyle.color.....	1372
5.15.62	chart.parallelAxis.axisLine.lineStyle.width .....	1373
5.15.63	chart.parallelAxis.axisLine.lineStyle.type.....	1374
	Erlaubte Werte .....	1374
5.15.64	chart.parallelAxis.axisLine.lineStyle.shadowBlur .....	1374
5.15.65	chart.parallelAxis.axisLine.lineStyle.shadowColor .....	1375
5.15.66	chart.parallelAxis.axisLine.lineStyle.shadowOffsetX.....	1375
5.15.67	chart.parallelAxis.axisLine.lineStyle.shadowOffsetY .....	1375
5.15.68	chart.parallelAxis.axisLine.lineStyle.opacity.....	1376
5.15.69	chart.parallelAxis.axisTick.object .....	1376
5.15.70	chart.parallelAxis.axisTick.show .....	1376
5.15.71	chart.parallelAxis.axisTick.alignWithLabel.....	1377
5.15.72	chart.parallelAxis.axisTick.interval .....	1377
5.15.73	chart.parallelAxis.axisTick.inside.....	1378
5.15.74	chart.parallelAxis.axisTick.length .....	1378
5.15.75	chart.parallelAxis.axisTick.lineStyle.object.....	1378
5.15.76	chart.parallelAxis.axisTick.lineStyle.color.....	1379
5.15.77	chart.parallelAxis.axisTick.lineStyle.width.....	1379
5.15.78	chart.parallelAxis.axisTick.lineStyle.type.....	1379
	Erlaubte Werte .....	1380
5.15.79	chart.parallelAxis.axisTick.lineStyle.shadowBlur .....	1380
5.15.80	chart.parallelAxis.axisTick.lineStyle.shadowColor .....	1380
5.15.81	chart.parallelAxis.axisTick.lineStyle.shadowOffsetX.....	1381

5.15.82	chart.parallelAxis.axisTick.lineStyle.shadowOffsetY .....	1381
5.15.83	chart.parallelAxis.axisTick.lineStyle.opacity .....	1381
5.15.84	chart.parallelAxis.axisLabel.object .....	1382
5.15.85	chart.parallelAxis.axisLabel.show .....	1382
5.15.86	chart.parallelAxis.axisLabel.interval .....	1382
5.15.87	chart.parallelAxis.axisLabel.inside .....	1383
5.15.88	chart.parallelAxis.axisLabel.rotate .....	1383
5.15.89	chart.parallelAxis.axisLabel.margin .....	1383
5.15.90	chart.parallelAxis.axisLabel.formatter .....	1384
5.15.91	chart.parallelAxis.axisLabel.showMinLabel .....	1384
5.15.92	chart.parallelAxis.axisLabel.showMaxLabel .....	1385
5.15.93	chart.parallelAxis.axisLabel.color .....	1385
5.15.94	chart.parallelAxis.axisLabel.fontStyle .....	1386
	Erlaubte Werte .....	1386
5.15.95	chart.parallelAxis.axisLabel.fontWeight .....	1386
	Erlaubte Werte .....	1387
5.15.96	chart.parallelAxis.axisLabel.fontFamily .....	1387
5.15.97	chart.parallelAxis.axisLabel.fontSize .....	1387
5.15.98	chart.parallelAxis.axisLabel.align .....	1388
	Erlaubte Werte .....	1388
5.15.99	chart.parallelAxis.axisLabel.verticalAlign .....	1388
	Erlaubte Werte .....	1389
5.15.100	chart.parallelAxis.axisLabel.lineHeight .....	1389
5.15.101	chart.parallelAxis.axisLabel.backgroundColor .....	1390
5.15.102	chart.parallelAxis.axisLabel.borderColor .....	1390
5.15.103	chart.parallelAxis.axisLabel.borderWidth .....	1391
5.15.104	chart.parallelAxis.axisLabel.borderRadius .....	1391
5.15.105	chart.parallelAxis.axisLabel.padding .....	1391
5.15.106	chart.parallelAxis.axisLabel.shadowColor .....	1392
5.15.107	chart.parallelAxis.axisLabel.shadowBlur .....	1392
5.15.108	chart.parallelAxis.axisLabel.shadowOffsetX .....	1393
5.15.109	chart.parallelAxis.axisLabel.shadowOffsetY .....	1393
5.15.110	chart.parallelAxis.axisLabel.width .....	1393
5.15.111	chart.parallelAxis.axisLabel.height .....	1394
5.15.112	chart.parallelAxis.axisLabel.textBorderColor .....	1394
5.15.113	chart.parallelAxis.axisLabel.textBorderWidth .....	1395

5.15.114	chart.parallelAxis.axisLabel.textShadowColor .....	1395
5.15.115	chart.parallelAxis.axisLabel.textShadowBlur.....	1395
5.15.116	chart.parallelAxis.axisLabel.textShadowOffsetX .....	1396
5.15.117	chart.parallelAxis.axisLabel.textShadowOffsetY.....	1396
5.15.118	chart.parallelAxis.axisLabel.rich.object.....	1396
<b>5.16</b>	<b>ECharts (4) Polar.....</b>	<b>1397</b>
5.16.1	chart.polar.object .....	1397
5.16.2	chart.polar.id.....	1398
5.16.3	chart.polar.zlevel .....	1398
5.16.4	chart.polar.z .....	1399
5.16.5	chart.polar.center .....	1399
5.16.6	chart.polar.radius .....	1400
5.16.7	chart.polar.tooltip.object.....	1400
5.16.8	chart.polar.tooltip.show.....	1401
5.16.9	chart.polar.tooltip.trigger .....	1401
	Erlaubte Werte .....	1401
5.16.10	chart.polar.tooltip.axisPointer.object .....	1402
5.16.11	chart.polar.tooltip.axisPointer.type .....	1402
	Erlaubte Werte .....	1403
5.16.12	chart.polar.tooltip.axisPointer.axis .....	1403
	Erlaubte Werte .....	1403
5.16.13	chart.polar.tooltip.axisPointer.snap.....	1403
5.16.14	chart.polar.tooltip.axisPointer.z .....	1404
5.16.15	chart.polar.tooltip.axisPointer.label.object.....	1404
5.16.16	chart.polar.tooltip.axisPointer.label.show .....	1404
5.16.17	chart.polar.tooltip.axisPointer.label.precision .....	1405
5.16.18	chart.polar.tooltip.axisPointer.label.formatter .....	1405
5.16.19	chart.polar.tooltip.axisPointer.label.margin .....	1407
5.16.20	chart.polar.tooltip.axisPointer.label.color .....	1407
5.16.21	chart.polar.tooltip.axisPointer.label.fontStyle .....	1407
5.16.22	chart.polar.tooltip.axisPointer.label.fontWeight.....	1408
	Erlaubte Werte .....	1408
5.16.23	chart.polar.tooltip.axisPointer.label.fontFamily .....	1408
5.16.24	chart.polar.tooltip.axisPointer.label.fontSize.....	1409
5.16.25	chart.polar.tooltip.axisPointer.label.lineHeight .....	1409
5.16.26	chart.polar.tooltip.axisPointer.label.width.....	1410



5.16.27	chart.polar.tooltip.axisPointer.label.height.....	1410
5.16.28	chart.polar.tooltip.axisPointer.label.textBorderColor .....	1410
5.16.29	chart.polar.tooltip.axisPointer.label.textBorderWidth .....	1411
5.16.30	chart.polar.tooltip.axisPointer.label.textShadowColor .....	1411
5.16.31	chart.polar.tooltip.axisPointer.label.textShadowBlur .....	1412
5.16.32	chart.polar.tooltip.axisPointer.label.textShadowOffsetX .....	1412
5.16.33	chart.polar.tooltip.axisPointer.label.textShadowOffsetY .....	1412
5.16.34	chart.polar.tooltip.axisPointer.label.padding .....	1413
5.16.35	chart.polar.tooltip.axisPointer.label.backgroundColor .....	1413
5.16.36	chart.polar.tooltip.axisPointer.label.borderColor .....	1414
5.16.37	chart.polar.tooltip.axisPointer.label.borderWidth .....	1414
5.16.38	chart.polar.tooltip.axisPointer.label.shadowBlur .....	1414
5.16.39	chart.polar.tooltip.axisPointer.label.shadowColor .....	1415
5.16.40	chart.polar.tooltip.axisPointer.label.shadowOffsetX .....	1415
5.16.41	chart.polar.tooltip.axisPointer.label.shadowOffsetY .....	1416
5.16.42	chart.polar.tooltip.axisPointer.lineStyle.object.....	1416
5.16.43	chart.polar.tooltip.axisPointer.lineStyle.color .....	1416
5.16.44	chart.polar.tooltip.axisPointer.lineStyle.width .....	1418
5.16.45	chart.polar.tooltip.axisPointer.lineStyle.type.....	1418
	Erlaubte Werte .....	1418
5.16.46	chart.polar.tooltip.axisPointer.lineStyle.shadowBlur .....	1418
5.16.47	chart.polar.tooltip.axisPointer.lineStyle.shadowColor .....	1419
5.16.48	chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetX.....	1419
5.16.49	chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetY .....	1420
5.16.50	chart.polar.tooltip.axisPointer.lineStyle.opacity.....	1420
5.16.51	chart.polar.tooltip.axisPointer.shadowStyle.object.....	1420
5.16.52	chart.polar.tooltip.axisPointer.shadowStyle.color.....	1421
5.16.53	chart.polar.tooltip.axisPointer.shadowStyle.shadowBlur .....	1422
5.16.54	chart.polar.tooltip.axisPointer.shadowStyle.shadowColor .....	1423
5.16.55	chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetX.....	1423
5.16.56	chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	1424
5.16.57	chart.polar.tooltip.axisPointer.shadowStyle.opacity.....	1424
5.16.58	chart.polar.tooltip.axisPointer.crossStyle.object .....	1424
5.16.59	chart.polar.tooltip.axisPointer.crossStyle.color .....	1425
5.16.60	chart.polar.tooltip.axisPointer.crossStyle.width.....	1426
5.16.61	chart.polar.tooltip.axisPointer.crossStyle.type .....	1427

	Erlaubte Werte .....	1427
5.16.62	chart.polar.tooltip.axisPointer.crossStyle.shadowBlur .....	1427
5.16.63	chart.polar.tooltip.axisPointer.crossStyle.shadowColor .....	1428
5.16.64	chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetX .....	1428
5.16.65	chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetY .....	1428
5.16.66	chart.polar.tooltip.axisPointer.crossStyle.opacity .....	1429
5.16.67	chart.polar.tooltip.position .....	1429
5.16.68	chart.polar.tooltip.formatter .....	1431
5.16.69	chart.polar.tooltip.backgroundColor .....	1433
5.16.70	chart.polar.tooltip.borderColor .....	1433
5.16.71	chart.polar.tooltip.borderWidth .....	1434
5.16.72	chart.polar.tooltip.padding .....	1434
5.16.73	chart.polar.tooltip.textStyle.object .....	1435
5.16.74	chart.polar.tooltip.textStyle.color .....	1435
5.16.75	chart.polar.tooltip.textStyle.fontStyle .....	1435
	Erlaubte Werte .....	1436
5.16.76	chart.polar.tooltip.textStyle.fontWeight .....	1436
	Erlaubte Werte .....	1436
5.16.77	chart.polar.tooltip.textStyle.fontFamily .....	1436
5.16.78	chart.polar.tooltip.textStyle.fontSize .....	1437
5.16.79	chart.polar.tooltip.textStyle.lineHeight .....	1437
5.16.80	chart.polar.tooltip.textStyle.width .....	1438
5.16.81	chart.polar.tooltip.textStyle.height .....	1438
5.16.82	chart.polar.tooltip.textStyle.textBorderColor .....	1439
5.16.83	chart.polar.tooltip.textStyle.textBorderWidth .....	1439
5.16.84	chart.polar.tooltip.textStyle.textShadowColor .....	1439
5.16.85	chart.polar.tooltip.textStyle.textShadowBlur .....	1440
5.16.86	chart.polar.tooltip.textStyle.textShadowOffsetX .....	1440
5.16.87	chart.polar.tooltip.textStyle.textShadowOffsetY .....	1440
5.16.88	chart.polar.tooltip.extraCssText .....	1441
5.17	ECharts (4) Radar .....	1441
5.17.1	chart.radar.object .....	1441
5.17.2	chart.radar.id .....	1442
5.17.3	chart.radar.zlevel .....	1442
5.17.4	chart.radar.z .....	1443
5.17.5	chart.radar.center .....	1443

5.17.6	chart.radar.radius .....	1444
5.17.7	chart.radar.startAngle .....	1444
5.17.8	chart.radar.name.object.....	1445
5.17.9	chart.radar.name.show .....	1445
5.17.10	chart.radar.name.formatter .....	1445
5.17.11	chart.radar.name.color.....	1446
5.17.12	chart.radar.name.fontStyle.....	1446
	Erlaubte Werte .....	1446
5.17.13	chart.radar.name.fontWeight .....	1447
	Erlaubte Werte .....	1447
5.17.14	chart.radar.name.fontFamily .....	1447
5.17.15	chart.radar.name.fontSize .....	1448
5.17.16	chart.radar.name.lineHeight.....	1448
5.17.17	chart.radar.name.backgroundColor.....	1448
5.17.18	chart.radar.name.borderColor.....	1449
5.17.19	chart.radar.name.borderWidth.....	1449
5.17.20	chart.radar.name.borderRadius .....	1450
5.17.21	chart.radar.name.padding .....	1450
5.17.22	chart.radar.name.shadowColor .....	1451
5.17.23	chart.radar.name.shadowBlur .....	1451
5.17.24	chart.radar.name.shadowOffsetX.....	1451
5.17.25	chart.radar.name.shadowOffsetY.....	1452
5.17.26	chart.radar.name.width .....	1452
	Erlaubte Werte .....	1453
5.17.27	chart.radar.name.height .....	1453
5.17.28	chart.radar.name.textBorderColor .....	1453
5.17.29	chart.radar.name.textBorderWidth .....	1453
5.17.30	chart.radar.name.textShadowColor.....	1454
5.17.31	chart.radar.name.textShadowBlur .....	1454
5.17.32	chart.radar.name.textShadowOffsetX.....	1454
5.17.33	chart.radar.name.textShadowOffsetY .....	1455
5.17.34	chart.radar.name.rich.object .....	1455
5.17.35	chart.radar.nameGap .....	1456
5.17.36	chart.radar.splitNumber.....	1457
5.17.37	chart.radar.shape.....	1457
	Erlaubte Werte .....	1457

5.17.38	chart.radar.scale .....	1457
5.17.39	chart.radar.silent .....	1458
5.17.40	chart.radar.triggerEvent .....	1458
5.17.41	chart.radar.axisLine.object .....	1459
5.17.42	chart.radar.axisLine.show .....	1459
5.17.43	chart.radar.axisLine.symbol.....	1460
	Erlaubte Werte .....	1460
5.17.44	chart.radar.axisLine.symbolSize .....	1460
5.17.45	chart.radar.axisLine.symbolOffset.....	1460
5.17.46	chart.radar.axisLine.lineStyle.object .....	1461
5.17.47	chart.radar.axisLine.lineStyle.color .....	1461
5.17.48	chart.radar.axisLine.lineStyle.width.....	1462
5.17.49	chart.radar.axisLine.lineStyle.type .....	1463
	Erlaubte Werte .....	1463
5.17.50	chart.radar.axisLine.lineStyle.shadowBlur .....	1463
5.17.51	chart.radar.axisLine.lineStyle.shadowColor .....	1464
5.17.52	chart.radar.axisLine.lineStyle.shadowOffsetX .....	1464
5.17.53	chart.radar.axisLine.lineStyle.shadowOffsetY .....	1464
5.17.54	chart.radar.axisLine.lineStyle.opacity .....	1465
5.17.55	chart.radar.splitLine.object.....	1465
5.17.56	chart.radar.splitLine.show .....	1465
5.17.57	chart.radar.splitLine.lineStyle.object .....	1466
5.17.58	chart.radar.splitLine.lineStyle.color .....	1466
5.17.59	chart.radar.splitLine.lineStyle.width .....	1467
5.17.60	chart.radar.splitLine.lineStyle.type .....	1467
	Erlaubte Werte .....	1467
5.17.61	chart.radar.splitLine.lineStyle.shadowBlur .....	1468
5.17.62	chart.radar.splitLine.lineStyle.shadowColor .....	1468
5.17.63	chart.radar.splitLine.lineStyle.shadowOffsetX .....	1468
5.17.64	chart.radar.splitLine.lineStyle.shadowOffsetY.....	1469
5.17.65	chart.radar.splitLine.lineStyle.opacity .....	1469
5.17.66	chart.radar.splitArea.object .....	1469
5.17.67	chart.radar.splitArea.show .....	1470
5.17.68	chart.radar.splitArea.areaStyle.object .....	1470
5.17.69	chart.radar.splitArea.areaStyle.color .....	1471
5.17.70	chart.radar.splitArea.areaStyle.shadowBlur.....	1471

5.17.71	chart.radar.splitArea.areaStyle.shadowColor.....	1471
5.17.72	chart.radar.splitArea.areaStyle.shadowOffsetX.....	1472
5.17.73	chart.radar.splitArea.areaStyle.shadowOffsetY.....	1472
5.17.74	chart.radar.splitArea.areaStyle.opacity.....	1472
5.17.75	chart.radar.indicator.object.....	1473
5.17.76	chart.radar.color.....	1473
<b>5.18</b>	<b>ECharts (4) Radius Axis.....</b>	<b>1474</b>
5.18.1	chart.radiusAxis.object.....	1474
5.18.2	chart.radiusAxis.id.....	1474
5.18.3	chart.radiusAxis.polarIndex.....	1474
5.18.4	chart.radiusAxis.type.....	1475
	Erlaubte Werte.....	1475
5.18.5	chart.radiusAxis.name.....	1476
5.18.6	chart.radiusAxis.nameLocation.....	1476
	Erlaubte Werte.....	1476
5.18.7	chart.radiusAxis.nameTextStyle.object.....	1476
5.18.8	chart.radiusAxis.nameTextStyle.color.....	1477
5.18.9	chart.radiusAxis.nameTextStyle.fontStyle.....	1477
	Erlaubte Werte.....	1477
5.18.10	chart.radiusAxis.nameTextStyle.fontWeight.....	1478
	Erlaubte Werte.....	1478
5.18.11	chart.radiusAxis.nameTextStyle.fontFamily.....	1478
5.18.12	chart.radiusAxis.nameTextStyle.fontSize.....	1479
5.18.13	chart.radiusAxis.nameTextStyle.align.....	1479
	Erlaubte Werte.....	1479
5.18.14	chart.radiusAxis.nameTextStyle.verticalAlign.....	1480
	Erlaubte Werte.....	1480
5.18.15	chart.radiusAxis.nameTextStyle.lineHeight.....	1480
5.18.16	chart.radiusAxis.nameTextStyle.backgroundColor.....	1481
5.18.17	chart.radiusAxis.nameTextStyle.borderColor.....	1481
5.18.18	chart.radiusAxis.nameTextStyle.borderWidth.....	1482
5.18.19	chart.radiusAxis.nameTextStyle.borderRadius.....	1482
5.18.20	chart.radiusAxis.nameTextStyle.padding.....	1483
5.18.21	chart.radiusAxis.nameTextStyle.shadowColor.....	1483
5.18.22	chart.radiusAxis.nameTextStyle.shadowBlur.....	1483
5.18.23	chart.radiusAxis.nameTextStyle.shadowOffsetX.....	1484

5.18.24	chart.radiusAxis.nameTextStyle.shadowOffsetY .....	1484
5.18.25	chart.radiusAxis.nameTextStyle.width .....	1484
5.18.26	chart.radiusAxis.nameTextStyle.height .....	1485
5.18.27	chart.radiusAxis.nameTextStyle.textBorderColor .....	1485
5.18.28	chart.radiusAxis.nameTextStyle.textBorderWidth .....	1486
5.18.29	chart.radiusAxis.nameTextStyle.textShadowColor .....	1486
5.18.30	chart.radiusAxis.nameTextStyle.textShadowBlur .....	1486
5.18.31	chart.radiusAxis.nameTextStyle.textShadowOffsetX .....	1487
5.18.32	chart.radiusAxis.nameTextStyle.textShadowOffsetY .....	1487
5.18.33	chart.radiusAxis.nameTextStyle.rich.object .....	1488
5.18.34	chart.radiusAxis.nameGap .....	1489
5.18.35	chart.radiusAxis.nameRotate .....	1489
5.18.36	chart.radiusAxis.inverse .....	1489
5.18.37	chart.radiusAxis.boundaryGap .....	1490
5.18.38	chart.radiusAxis.min .....	1490
5.18.39	chart.radiusAxis.max .....	1491
5.18.40	chart.radiusAxis.scale .....	1491
5.18.41	chart.radiusAxis.splitNumber .....	1492
5.18.42	chart.radiusAxis.minInterval .....	1492
5.18.43	chart.radiusAxis.interval .....	1492
5.18.44	chart.radiusAxis.logBase .....	1493
5.18.45	chart.radiusAxis.silent .....	1493
5.18.46	chart.radiusAxis.triggerEvent .....	1494
5.18.47	chart.radiusAxis.axisLine.object .....	1494
5.18.48	chart.radiusAxis.axisLine.show .....	1495
5.18.49	chart.radiusAxis.axisLine.symbol .....	1495
	Erlaubte Werte .....	1495
5.18.50	chart.radiusAxis.axisLine.symbolSize .....	1496
5.18.51	chart.radiusAxis.axisLine.symbolOffset .....	1496
5.18.52	chart.radiusAxis.axisLine.lineStyle.object .....	1496
5.18.53	chart.radiusAxis.axisLine.lineStyle.color .....	1497
5.18.54	chart.radiusAxis.axisLine.lineStyle.width .....	1498
5.18.55	chart.radiusAxis.axisLine.lineStyle.type .....	1499
	Erlaubte Werte .....	1499
5.18.56	chart.radiusAxis.axisLine.lineStyle.shadowBlur .....	1499
5.18.57	chart.radiusAxis.axisLine.lineStyle.shadowColor .....	1500

5.18.58	chart.radiusAxis.axisLine.lineStyle.shadowOffsetX .....	1500
5.18.59	chart.radiusAxis.axisLine.lineStyle.shadowOffsetY .....	1500
5.18.60	chart.radiusAxis.axisLine.lineStyle.opacity .....	1501
5.18.61	chart.radiusAxis.axisTick.object .....	1501
5.18.62	chart.radiusAxis.axisTick.show .....	1501
5.18.63	chart.radiusAxis.axisTick.alignWithLabel .....	1502
5.18.64	chart.radiusAxis.axisTick.interval .....	1502
5.18.65	chart.radiusAxis.axisTick.inside .....	1503
5.18.66	chart.radiusAxis.axisTick.length .....	1503
5.18.67	chart.radiusAxis.axisTick.lineStyle.object .....	1503
5.18.68	chart.radiusAxis.axisTick.lineStyle.color .....	1504
5.18.69	chart.radiusAxis.axisTick.lineStyle.width .....	1504
5.18.70	chart.radiusAxis.axisTick.lineStyle.type .....	1504
	Erlaubte Werte .....	1505
5.18.71	chart.radiusAxis.axisTick.lineStyle.shadowBlur .....	1505
5.18.72	chart.radiusAxis.axisTick.lineStyle.shadowColor .....	1505
5.18.73	chart.radiusAxis.axisTick.lineStyle.shadowOffsetX .....	1506
5.18.74	chart.radiusAxis.axisTick.lineStyle.shadowOffsetY .....	1506
5.18.75	chart.radiusAxis.axisTick.lineStyle.opacity .....	1506
5.18.76	chart.radiusAxis.axisLabel.object .....	1507
5.18.77	chart.radiusAxis.axisLabel.show .....	1507
5.18.78	chart.radiusAxis.axisLabel.interval .....	1507
5.18.79	chart.radiusAxis.axisLabel.inside .....	1508
5.18.80	chart.radiusAxis.axisLabel.rotate .....	1508
5.18.81	chart.radiusAxis.axisLabel.margin .....	1508
5.18.82	chart.radiusAxis.axisLabel.formatter .....	1509
5.18.83	chart.radiusAxis.axisLabel.showMinLabel .....	1509
5.18.84	chart.radiusAxis.axisLabel.showMaxLabel .....	1510
5.18.85	chart.radiusAxis.axisLabel.color .....	1510
5.18.86	chart.radiusAxis.axisLabel.fontStyle .....	1511
	Erlaubte Werte .....	1511
5.18.87	chart.radiusAxis.axisLabel.fontWeight .....	1511
	Erlaubte Werte .....	1512
5.18.88	chart.radiusAxis.axisLabel.fontFamily .....	1512
5.18.89	chart.radiusAxis.axisLabel.fontSize .....	1512
5.18.90	chart.radiusAxis.axisLabel.align .....	1513

	Erlaubte Werte .....	1513
5.18.91	chart.radiusAxis.axisLabel.verticalAlign .....	1513
	Erlaubte Werte .....	1514
5.18.92	chart.radiusAxis.axisLabel.lineHeight .....	1514
5.18.93	chart.radiusAxis.axisLabel.backgroundColor.....	1515
5.18.94	chart.radiusAxis.axisLabel.borderColor .....	1515
5.18.95	chart.radiusAxis.axisLabel.borderWidth.....	1516
5.18.96	chart.radiusAxis.axisLabel.borderRadius .....	1516
5.18.97	chart.radiusAxis.axisLabel.padding.....	1516
5.18.98	chart.radiusAxis.axisLabel.shadowColor.....	1517
5.18.99	chart.radiusAxis.axisLabel.shadowBlur.....	1517
5.18.100	chart.radiusAxis.axisLabel.shadowOffsetX.....	1518
5.18.101	chart.radiusAxis.axisLabel.shadowOffsetY.....	1518
5.18.102	chart.radiusAxis.axisLabel.width .....	1518
5.18.103	chart.radiusAxis.axisLabel.height .....	1519
5.18.104	chart.radiusAxis.axisLabel.textBorderColor.....	1519
5.18.105	chart.radiusAxis.axisLabel.textBorderWidth .....	1520
5.18.106	chart.radiusAxis.axisLabel.textShadowColor.....	1520
5.18.107	chart.radiusAxis.axisLabel.textShadowBlur.....	1520
5.18.108	chart.radiusAxis.axisLabel.textShadowOffsetX.....	1521
5.18.109	chart.radiusAxis.axisLabel.textShadowOffsetY.....	1521
5.18.110	chart.radiusAxis.axisLabel.rich.object.....	1521
5.18.111	chart.radiusAxis.splitLine.object.....	1522
5.18.112	chart.radiusAxis.splitLine.show .....	1523
5.18.113	chart.radiusAxis.splitLine.interval .....	1523
5.18.114	chart.radiusAxis.splitLine.lineStyle.object .....	1523
5.18.115	chart.radiusAxis.splitLine.lineStyle.color .....	1524
5.18.116	chart.radiusAxis.splitLine.lineStyle.width .....	1524
5.18.117	chart.radiusAxis.splitLine.lineStyle.type .....	1525
	Erlaubte Werte .....	1525
5.18.118	chart.radiusAxis.splitLine.lineStyle.shadowBlur .....	1525
5.18.119	chart.radiusAxis.splitLine.lineStyle.shadowColor .....	1526
5.18.120	chart.radiusAxis.splitLine.lineStyle.shadowOffsetX .....	1526
5.18.121	chart.radiusAxis.splitLine.lineStyle.shadowOffsetY .....	1526
5.18.122	chart.radiusAxis.splitLine.lineStyle.opacity .....	1527
5.18.123	chart.radiusAxis.splitArea.object .....	1527



5.18.124	chart.radiusAxis.splitArea.interval	1527
5.18.125	chart.radiusAxis.splitArea.show	1528
5.18.126	chart.radiusAxis.splitArea.areaStyle.object	1528
5.18.127	chart.radiusAxis.splitArea.areaStyle.color	1528
5.18.128	chart.radiusAxis.splitArea.areaStyle.shadowBlur	1529
5.18.129	chart.radiusAxis.splitArea.areaStyle.shadowColor	1529
5.18.130	chart.radiusAxis.splitArea.areaStyle.shadowOffsetX	1530
5.18.131	chart.radiusAxis.splitArea.areaStyle.shadowOffsetY	1530
5.18.132	chart.radiusAxis.splitArea.areaStyle.opacity	1530
5.18.133	chart.radiusAxis.axisPointer.object	1531
5.18.134	chart.radiusAxis.axisPointer.show	1531
5.18.135	chart.radiusAxis.axisPointer.type	1532
	Erlaubte Werte	1532
5.18.136	chart.radiusAxis.axisPointer.snap	1532
5.18.137	chart.radiusAxis.axisPointer.z	1532
5.18.138	chart.radiusAxis.axisPointer.label.object	1533
5.18.139	chart.radiusAxis.axisPointer.label.show	1533
5.18.140	chart.radiusAxis.axisPointer.label.precision	1534
5.18.141	chart.radiusAxis.axisPointer.label.formatter	1534
5.18.142	chart.radiusAxis.axisPointer.label.margin	1535
5.18.143	chart.radiusAxis.axisPointer.label.color	1536
5.18.144	chart.radiusAxis.axisPointer.label.fontStyle	1536
	Erlaubte Werte	1536
5.18.145	chart.radiusAxis.axisPointer.label.fontWeight	1537
	Erlaubte Werte	1537
5.18.146	chart.radiusAxis.axisPointer.label.fontFamily	1537
5.18.147	chart.radiusAxis.axisPointer.label.fontSize	1538
5.18.148	chart.radiusAxis.axisPointer.label.lineHeight	1538
5.18.149	chart.radiusAxis.axisPointer.label.width	1539
5.18.150	chart.radiusAxis.axisPointer.label.height	1539
5.18.151	chart.radiusAxis.axisPointer.label.textBorderColor	1539
5.18.152	chart.radiusAxis.axisPointer.label.textBorderWidth	1540
5.18.153	chart.radiusAxis.axisPointer.label.textShadowColor	1540
5.18.154	chart.radiusAxis.axisPointer.label.textShadowBlur	1541
5.18.155	chart.radiusAxis.axisPointer.label.textShadowOffsetX	1541
5.18.156	chart.radiusAxis.axisPointer.label.textShadowOffsetY	1541

5.18.157	chart.radiusAxis.axisPointer.label.padding .....	1542
5.18.158	chart.radiusAxis.axisPointer.label.backgroundColor .....	1542
5.18.159	chart.radiusAxis.axisPointer.label.borderColor .....	1543
5.18.160	chart.radiusAxis.axisPointer.label.borderWidth .....	1543
5.18.161	chart.radiusAxis.axisPointer.label.shadowBlur .....	1543
5.18.162	chart.radiusAxis.axisPointer.label.shadowColor .....	1544
5.18.163	chart.radiusAxis.axisPointer.label.shadowOffsetX .....	1544
5.18.164	chart.radiusAxis.axisPointer.label.shadowOffsetY .....	1545
5.18.165	chart.radiusAxis.axisPointer.lineStyle.object.....	1545
5.18.166	chart.radiusAxis.axisPointer.lineStyle.color.....	1545
5.18.167	chart.radiusAxis.axisPointer.lineStyle.width .....	1547
5.18.168	chart.radiusAxis.axisPointer.lineStyle.type.....	1547
	Erlaubte Werte .....	1547
5.18.169	chart.radiusAxis.axisPointer.lineStyle.shadowBlur .....	1547
5.18.170	chart.radiusAxis.axisPointer.lineStyle.shadowColor.....	1548
5.18.171	chart.radiusAxis.axisPointer.lineStyle.shadowOffsetX.....	1548
5.18.172	chart.radiusAxis.axisPointer.lineStyle.shadowOffsetY .....	1549
5.18.173	chart.radiusAxis.axisPointer.lineStyle.opacity.....	1549
5.18.174	chart.radiusAxis.axisPointer.shadowStyle.object.....	1549
5.18.175	chart.radiusAxis.axisPointer.shadowStyle.color.....	1550
5.18.176	chart.radiusAxis.axisPointer.shadowStyle.shadowBlur .....	1551
5.18.177	chart.radiusAxis.axisPointer.shadowStyle.shadowColor .....	1552
5.18.178	chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetX.....	1552
5.18.179	chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetY .....	1553
5.18.180	chart.radiusAxis.axisPointer.shadowStyle.opacity.....	1553
5.18.181	chart.radiusAxis.axisPointer.triggerTooltip .....	1553
5.18.182	chart.radiusAxis.axisPointer.value.....	1554
5.18.183	chart.radiusAxis.axisPointer.status .....	1554
	Erlaubte Werte .....	1554
5.18.184	chart.radiusAxis.axisPointer.handle.object .....	1554
5.18.185	chart.radiusAxis.axisPointer.handle.show .....	1555
5.18.186	chart.radiusAxis.axisPointer.handle.icon .....	1555
5.18.187	chart.radiusAxis.axisPointer.handle.size.....	1556
5.18.188	chart.radiusAxis.axisPointer.handle.margin .....	1556
5.18.189	chart.radiusAxis.axisPointer.handle.color .....	1556
5.18.190	chart.radiusAxis.axisPointer.handle.throttle .....	1557

5.18.191	chart.radiusAxis.axisPointer.handle.shadowBlur .....	1557
5.18.192	chart.radiusAxis.axisPointer.handle.shadowColor .....	1558
5.18.193	chart.radiusAxis.axisPointer.handle.shadowOffsetX .....	1558
5.18.194	chart.radiusAxis.axisPointer.handle.shadowOffsetY .....	1558
5.18.195	chart.radiusAxis.zlevel .....	1559
5.18.196	chart.radiusAxis.z .....	1559
5.19	<b>ECharts (4) Series .....</b>	<b>1560</b>
5.19.1	chart.series.i[].activeOpacity .....	1560
5.19.2	chart.series.i[].animation .....	1560
5.19.3	chart.series.i[].animationDelay .....	1560
5.19.4	chart.series.i[].animationDelayUpdate .....	1561
5.19.5	chart.series.i[].animationDuration .....	1562
5.19.6	chart.series.i[].animationDurationUpdate .....	1562
5.19.7	chart.series.i[].animationEasing .....	1563
	Erlaubte Werte .....	1563
5.19.8	chart.series.i[].animationEasingUpdate .....	1564
	Erlaubte Werte .....	1564
5.19.9	chart.series.i[].animationEasingUpdate.animationDelay .....	1565
5.19.10	chart.series.i[].animationEasingUpdate.animationDelayUpdate .....	1565
5.19.11	chart.series.i[].animationEasingUpdate.object .....	1566
	Erlaubte Werte .....	1566
5.19.12	chart.series.i[].animationThreshold .....	1567
5.19.13	chart.series.i[].animationType .....	1567
	Erlaubte Werte .....	1567
5.19.14	chart.series.i[].areaStyle.color .....	1568
5.19.15	chart.series.i[].areaStyle.object .....	1569
5.19.16	chart.series.i[].areaStyle.opacity .....	1570
5.19.17	chart.series.i[].areaStyle.origin .....	1570
	Erlaubte Werte .....	1570
5.19.18	chart.series.i[].areaStyle.shadowBlur .....	1571
5.19.19	chart.series.i[].areaStyle.shadowColor .....	1571
5.19.20	chart.series.i[].areaStyle.shadowOffsetX .....	1571
5.19.21	chart.series.i[].areaStyle.shadowOffsetY .....	1572
5.19.22	chart.series.i[].aspectScale .....	1572
5.19.23	chart.series.i[].avoidLabelOverlap .....	1573
5.19.24	chart.series.i[].axisLabel.backgroundColor .....	1573

5.19.25	chart.series.i[].axisLabel.borderColor .....	1574
5.19.26	chart.series.i[].axisLabel.borderRadius .....	1574
5.19.27	chart.series.i[].axisLabel.borderWidth.....	1574
5.19.28	chart.series.i[].axisLabel.color .....	1575
5.19.29	chart.series.i[].axisLabel.fontFamily.....	1575
5.19.30	chart.series.i[].axisLabel.fontSize .....	1575
5.19.31	chart.series.i[].axisLabel.fontStyle.....	1576
	Erlaubte Werte .....	1576
5.19.32	chart.series.i[].axisLabel.fontWeight .....	1576
	Erlaubte Werte .....	1577
5.19.33	chart.series.i[].axisLabel.formatter.....	1577
5.19.34	chart.series.i[].axisLabel.height .....	1577
5.19.35	chart.series.i[].axisLabel.lineHeight .....	1578
5.19.36	chart.series.i[].axisLabel.object .....	1578
5.19.37	chart.series.i[].axisLabel.padding.....	1579
5.19.38	chart.series.i[].axisLabel.rich.object.....	1579
5.19.39	chart.series.i[].axisLabel.shadowBlur.....	1580
5.19.40	chart.series.i[].axisLabel.shadowColor.....	1581
5.19.41	chart.series.i[].axisLabel.shadowOffsetX.....	1581
5.19.42	chart.series.i[].axisLabel.shadowOffsetY.....	1581
5.19.43	chart.series.i[].axisLabel.show .....	1582
5.19.44	chart.series.i[].axisLabel.textBorderColor.....	1582
5.19.45	chart.series.i[].axisLabel.textBorderWidth.....	1582
5.19.46	chart.series.i[].axisLabel.textShadowBlur.....	1583
5.19.47	chart.series.i[].axisLabel.textShadowColor.....	1583
5.19.48	chart.series.i[].axisLabel.textShadowOffsetX.....	1583
5.19.49	chart.series.i[].axisLabel.textShadowOffsetY.....	1584
5.19.50	chart.series.i[].axisLabel.width .....	1584
5.19.51	chart.series.i[].axisLine.lineStyle.color.....	1585
5.19.52	chart.series.i[].axisLine.lineStyle.object.....	1585
5.19.53	chart.series.i[].axisLine.lineStyle.opacity.....	1585
5.19.54	chart.series.i[].axisLine.lineStyle.shadowBlur .....	1586
5.19.55	chart.series.i[].axisLine.lineStyle.shadowColor .....	1586
5.19.56	chart.series.i[].axisLine.lineStyle.shadowOffsetX .....	1587
5.19.57	chart.series.i[].axisLine.lineStyle.shadowOffsetY .....	1587
5.19.58	chart.series.i[].axisLine.lineStyle.width.....	1587

5.19.59	chart.series.i[].axisLine.object .....	1588
5.19.60	chart.series.i[].axisLine.show .....	1588
5.19.61	chart.series.i[].axisTick.length .....	1588
5.19.62	chart.series.i[].axisTick.lineStyle.color .....	1589
5.19.63	chart.series.i[].axisTick.lineStyle.object .....	1590
5.19.64	chart.series.i[].axisTick.lineStyle.opacity .....	1591
5.19.65	chart.series.i[].axisTick.lineStyle.shadowBlur .....	1591
5.19.66	chart.series.i[].axisTick.lineStyle.shadowColor .....	1592
5.19.67	chart.series.i[].axisTick.lineStyle.shadowOffsetX .....	1592
5.19.68	chart.series.i[].axisTick.lineStyle.shadowOffsetY .....	1592
5.19.69	chart.series.i[].axisTick.lineStyle.type .....	1593
	Erlaubte Werte .....	1593
5.19.70	chart.series.i[].axisTick.lineStyle.width .....	1593
5.19.71	chart.series.i[].axisTick.object.....	1593
5.19.72	chart.series.i[].axisTick.show .....	1594
5.19.73	chart.series.i[].axisTick.splitNumber .....	1594
5.19.74	chart.series.i[].barCategoryGap .....	1594
5.19.75	chart.series.i[].barGap .....	1595
5.19.76	chart.series.i[].barMaxWidth .....	1595
5.19.77	chart.series.i[].barMinHeight.....	1596
5.19.78	chart.series.i[].barMinWidth .....	1596
	Erlaubte Werte .....	1597
5.19.79	chart.series.i[].barWidth .....	1597
5.19.80	chart.series.i[].blurSize .....	1597
5.19.81	chart.series.i[].bottom .....	1597
	Erlaubte Werte .....	1598
5.19.82	chart.series.i[].boundaryGap.....	1598
	Erlaubte Werte .....	1598
5.19.83	chart.series.i[].boundingCoords .....	1598
5.19.84	chart.series.i[].boxWidth .....	1599
5.19.85	chart.series.i[].breadcrumb.bottom .....	1599
	Erlaubte Werte .....	1600
5.19.86	chart.series.i[].breadcrumb.emphasis.itemStyle.borderColor .....	1600
5.19.87	chart.series.i[].breadcrumb.emphasis.itemStyle.borderType .....	1600
	Erlaubte Werte .....	1601
5.19.88	chart.series.i[].breadcrumb.emphasis.itemStyle.borderWidth .....	1601

5.19.89	chart.series.i[].breadcrumb.emphasis.itemStyle.color .....	1601
5.19.90	chart.series.i[].breadcrumb.emphasis.itemStyle.object .....	1602
5.19.91	chart.series.i[].breadcrumb.emphasis.itemStyle.opacity .....	1603
5.19.92	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowBlur .....	1603
5.19.93	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowColor .....	1604
5.19.94	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetX .....	1604
5.19.95	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetY .....	1604
5.19.96	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.align .....	1605
	Erlaubte Werte .....	1605
5.19.97	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.backgroundColor .....	1605
5.19.98	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderColor .....	1606
5.19.99	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderRadius .....	1606
5.19.100	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderWidth .....	1607
5.19.101	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.color .....	1607
5.19.102	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontFamily.....	1607
5.19.103	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontSize .....	1608
5.19.104	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontStyle .....	1608
	Erlaubte Werte .....	1608
5.19.105	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontWeight .....	1609
	Erlaubte Werte .....	1609
5.19.106	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.height .....	1609
5.19.107	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.lineHeight .....	1610
5.19.108	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.object .....	1610
5.19.109	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.padding.....	1611
5.19.110	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.rich.object.....	1611
5.19.111	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowBlur.....	1612
5.19.112	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowColor .....	1613
5.19.113	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetX .....	1613
5.19.114	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetY.....	1613
5.19.115	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textBorderColor.....	1614
5.19.116	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textBorderWidth.....	1614
5.19.117	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowBlur.....	1615
5.19.118	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowColor .....	1615
5.19.119	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetX .....	1615
5.19.120	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetY.....	1616
5.19.121	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.verticalAlign.....	1616

	Erlaubte Werte .....	1617
5.19.122	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.width .....	1617
	Erlaubte Werte .....	1617
5.19.123	chart.series.i[].breadcrumb.emphasis.object .....	1618
5.19.124	chart.series.i[].breadcrumb.emptyItemWidth .....	1618
5.19.125	chart.series.i[].breadcrumb.height .....	1618
5.19.126	chart.series.i[].breadcrumb.itemStyle.borderColor .....	1619
5.19.127	chart.series.i[].breadcrumb.itemStyle.borderType .....	1619
	Erlaubte Werte .....	1619
5.19.128	chart.series.i[].breadcrumb.itemStyle.borderWidth .....	1619
5.19.129	chart.series.i[].breadcrumb.itemStyle.color .....	1620
5.19.130	chart.series.i[].breadcrumb.itemStyle.object .....	1621
5.19.131	chart.series.i[].breadcrumb.itemStyle.opacity .....	1622
5.19.132	chart.series.i[].breadcrumb.itemStyle.shadowBlur.....	1622
5.19.133	chart.series.i[].breadcrumb.itemStyle.shadowColor .....	1623
5.19.134	chart.series.i[].breadcrumb.itemStyle.shadowOffsetX .....	1623
5.19.135	chart.series.i[].breadcrumb.itemStyle.shadowOffsetY.....	1623
5.19.136	chart.series.i[].breadcrumb.itemStyle.textStyle.align.....	1624
	Erlaubte Werte .....	1624
5.19.137	chart.series.i[].breadcrumb.itemStyle.textStyle.backgroundColor .....	1624
5.19.138	chart.series.i[].breadcrumb.itemStyle.textStyle.borderColor .....	1625
5.19.139	chart.series.i[].breadcrumb.itemStyle.textStyle.borderRadius .....	1625
5.19.140	chart.series.i[].breadcrumb.itemStyle.textStyle.borderWidth .....	1626
5.19.141	chart.series.i[].breadcrumb.itemStyle.textStyle.color .....	1626
5.19.142	chart.series.i[].breadcrumb.itemStyle.textStyle.fontFamily.....	1626
5.19.143	chart.series.i[].breadcrumb.itemStyle.textStyle.fontSize .....	1627
5.19.144	chart.series.i[].breadcrumb.itemStyle.textStyle.fontStyle .....	1627
	Erlaubte Werte .....	1627
5.19.145	chart.series.i[].breadcrumb.itemStyle.textStyle.fontWeight .....	1628
	Erlaubte Werte .....	1628
5.19.146	chart.series.i[].breadcrumb.itemStyle.textStyle.height .....	1628
5.19.147	chart.series.i[].breadcrumb.itemStyle.textStyle.lineHeight .....	1629
5.19.148	chart.series.i[].breadcrumb.itemStyle.textStyle.object .....	1629
5.19.149	chart.series.i[].breadcrumb.itemStyle.textStyle.padding.....	1630
5.19.150	chart.series.i[].breadcrumb.itemStyle.textStyle.rich.object.....	1630
5.19.151	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowBlur.....	1631

5.19.152	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowColor.....	1632
5.19.153	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetX.....	1632
5.19.154	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetY.....	1632
5.19.155	chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderColor.....	1633
5.19.156	chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderWidth.....	1633
5.19.157	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowBlur.....	1633
5.19.158	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowColor.....	1634
5.19.159	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetX.....	1634
5.19.160	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetY.....	1634
5.19.161	chart.series.i[].breadcrumb.itemStyle.textStyle.verticalAlign.....	1635
	Erlaubte Werte .....	1635
5.19.162	chart.series.i[].breadcrumb.itemStyle.textStyle.width .....	1635
	Erlaubte Werte .....	1636
5.19.163	chart.series.i[].breadcrumb.left .....	1636
	Erlaubte Werte .....	1636
5.19.164	chart.series.i[].breadcrumb.object .....	1637
5.19.165	chart.series.i[].breadcrumb.right.....	1637
	Erlaubte Werte .....	1637
5.19.166	chart.series.i[].breadcrumb.show.....	1638
5.19.167	chart.series.i[].breadcrumb.top .....	1638
	Erlaubte Werte .....	1638
5.19.168	chart.series.i[].calendarIndex.....	1639
5.19.169	chart.series.i[].categories.emphasis.itemStyle.borderColor .....	1639
5.19.170	chart.series.i[].categories.emphasis.itemStyle.borderType .....	1639
	Erlaubte Werte .....	1640
5.19.171	chart.series.i[].categories.emphasis.itemStyle.borderWidth.....	1640
5.19.172	chart.series.i[].categories.emphasis.itemStyle.color .....	1640
5.19.173	chart.series.i[].categories.emphasis.itemStyle.object .....	1641
5.19.174	chart.series.i[].categories.emphasis.itemStyle.opacity .....	1642
5.19.175	chart.series.i[].categories.emphasis.itemStyle.shadowBlur.....	1642
5.19.176	chart.series.i[].categories.emphasis.itemStyle.shadowColor.....	1643
5.19.177	chart.series.i[].categories.emphasis.itemStyle.shadowOffsetX.....	1643
5.19.178	chart.series.i[].categories.emphasis.itemStyle.shadowOffsetY.....	1643
5.19.179	chart.series.i[].categories.emphasis.label.align .....	1644
	Erlaubte Werte .....	1644
5.19.180	chart.series.i[].categories.emphasis.label.backgroundColor .....	1644



5.19.181	chart.series.i[].categories.emphasis.label.borderColor .....	1645
5.19.182	chart.series.i[].categories.emphasis.label.borderRadius .....	1645
5.19.183	chart.series.i[].categories.emphasis.label.borderWidth .....	1646
5.19.184	chart.series.i[].categories.emphasis.label.color .....	1646
5.19.185	chart.series.i[].categories.emphasis.label.distance .....	1646
5.19.186	chart.series.i[].categories.emphasis.label.fontFamily.....	1647
5.19.187	chart.series.i[].categories.emphasis.label.fontSize .....	1647
5.19.188	chart.series.i[].categories.emphasis.label.fontStyle .....	1648
	Erlaubte Werte .....	1648
5.19.189	chart.series.i[].categories.emphasis.label.fontWeight .....	1648
	Erlaubte Werte .....	1648
5.19.190	chart.series.i[].categories.emphasis.label.formatter .....	1649
5.19.191	chart.series.i[].categories.emphasis.label.height .....	1650
5.19.192	chart.series.i[].categories.emphasis.label.lineHeight .....	1650
5.19.193	chart.series.i[].categories.emphasis.label.object .....	1651
5.19.194	chart.series.i[].categories.emphasis.label.offset .....	1651
5.19.195	chart.series.i[].categories.emphasis.label.padding.....	1652
5.19.196	chart.series.i[].categories.emphasis.label.position .....	1652
	Erlaubte Werte .....	1653
5.19.197	chart.series.i[].categories.emphasis.label.rich.object.....	1653
5.19.198	chart.series.i[].categories.emphasis.label.rotate .....	1654
5.19.199	chart.series.i[].categories.emphasis.label.shadowBlur.....	1655
5.19.200	chart.series.i[].categories.emphasis.label.shadowColor .....	1655
5.19.201	chart.series.i[].categories.emphasis.label.shadowOffsetX .....	1655
5.19.202	chart.series.i[].categories.emphasis.label.shadowOffsetY.....	1656
5.19.203	chart.series.i[].categories.emphasis.label.show.....	1656
5.19.204	chart.series.i[].categories.emphasis.label.textBorderColor.....	1656
5.19.205	chart.series.i[].categories.emphasis.label.textBorderWidth.....	1657
5.19.206	chart.series.i[].categories.emphasis.label.textShadowBlur.....	1657
5.19.207	chart.series.i[].categories.emphasis.label.textShadowColor .....	1657
5.19.208	chart.series.i[].categories.emphasis.label.textShadowOffsetX .....	1658
5.19.209	chart.series.i[].categories.emphasis.label.textShadowOffsetY.....	1658
5.19.210	chart.series.i[].categories.emphasis.label.verticalAlign.....	1659
	Erlaubte Werte .....	1659
5.19.211	chart.series.i[].categories.emphasis.label.width .....	1659
5.19.212	chart.series.i[].categories.emphasis.object .....	1660

5.19.213	chart.series.i[].categories.itemStyle.borderColor.....	1660
5.19.214	chart.series.i[].categories.itemStyle.borderColor	1660
	Erlaubte Werte .....	1661
5.19.215	chart.series.i[].categories.itemStyle.borderWidth.....	1661
5.19.216	chart.series.i[].categories.itemStyle.color .....	1661
5.19.217	chart.series.i[].categories.itemStyle.object .....	1663
5.19.218	chart.series.i[].categories.itemStyle.opacity.....	1663
5.19.219	chart.series.i[].categories.itemStyle.shadowBlur.....	1663
5.19.220	chart.series.i[].categories.itemStyle.shadowColor.....	1664
5.19.221	chart.series.i[].categories.itemStyle.shadowOffsetX.....	1664
5.19.222	chart.series.i[].categories.itemStyle.shadowOffsetY.....	1664
5.19.223	chart.series.i[].categories.label.align.....	1665
	Erlaubte Werte .....	1665
5.19.224	chart.series.i[].categories.label.backgroundColor .....	1665
5.19.225	chart.series.i[].categories.label.borderColor .....	1666
5.19.226	chart.series.i[].categories.label.borderRadius .....	1667
5.19.227	chart.series.i[].categories.label.borderWidth .....	1667
5.19.228	chart.series.i[].categories.label.color .....	1667
5.19.229	chart.series.i[].categories.label.distance.....	1668
5.19.230	chart.series.i[].categories.label.fontFamily.....	1668
5.19.231	chart.series.i[].categories.label.fontSize .....	1668
5.19.232	chart.series.i[].categories.label.fontStyle.....	1669
	Erlaubte Werte .....	1669
5.19.233	chart.series.i[].categories.label.fontWeight .....	1669
	Erlaubte Werte .....	1670
5.19.234	chart.series.i[].categories.label.formatter.....	1670
5.19.235	chart.series.i[].categories.label.height .....	1671
5.19.236	chart.series.i[].categories.label.lineHeight .....	1671
5.19.237	chart.series.i[].categories.label.object .....	1672
5.19.238	chart.series.i[].categories.label.offset .....	1672
5.19.239	chart.series.i[].categories.label.padding.....	1673
5.19.240	chart.series.i[].categories.label.position .....	1673
	Erlaubte Werte .....	1674
5.19.241	chart.series.i[].categories.label.rich.object.....	1674
5.19.242	chart.series.i[].categories.label.rotate.....	1675
5.19.243	chart.series.i[].categories.label.shadowBlur.....	1676

5.19.244	chart.series.i[].categories.label.shadowColor.....	1676
5.19.245	chart.series.i[].categories.label.shadowOffsetX.....	1676
5.19.246	chart.series.i[].categories.label.shadowOffsetY.....	1677
5.19.247	chart.series.i[].categories.label.show.....	1677
5.19.248	chart.series.i[].categories.label.textBorderColor.....	1677
5.19.249	chart.series.i[].categories.label.textBorderWidth.....	1678
5.19.250	chart.series.i[].categories.label.textShadowBlur.....	1678
5.19.251	chart.series.i[].categories.label.textShadowColor.....	1678
5.19.252	chart.series.i[].categories.label.textShadowOffsetX.....	1679
5.19.253	chart.series.i[].categories.label.textShadowOffsetY.....	1679
5.19.254	chart.series.i[].categories.label.verticalAlign.....	1680
	Erlaubte Werte .....	1680
5.19.255	chart.series.i[].categories.label.width.....	1680
5.19.256	chart.series.i[].categories.name.....	1681
5.19.257	chart.series.i[].categories.object.....	1681
5.19.258	chart.series.i[].categories.symbol.....	1681
	Erlaubte Werte .....	1682
5.19.259	chart.series.i[].categories.symbolKeepAspect.....	1682
5.19.260	chart.series.i[].categories.symbolOffset.....	1683
5.19.261	chart.series.i[].categories.symbolRotate.....	1683
5.19.262	chart.series.i[].categories.symbolSize.....	1683
5.19.263	chart.series.i[].center.....	1684
5.19.264	chart.series.i[].childrenVisibleMin.....	1684
5.19.265	chart.series.i[].circular.object.....	1685
5.19.266	chart.series.i[].circular.rotateLabel.....	1685
5.19.267	chart.series.i[].clipOverflow.....	1686
5.19.268	chart.series.i[].clockwise.....	1686
5.19.269	chart.series.i[].colorAlpha.....	1686
5.19.270	chart.series.i[].colorMappingBy.....	1687
	Erlaubte Werte .....	1688
5.19.271	chart.series.i[].colorSaturation.....	1688
5.19.272	chart.series.i[].connectNulls.....	1688
5.19.273	chart.series.i[].coordinateSystem.....	1689
	Erlaubte Werte .....	1689
5.19.274	chart.series.i[].cursor.....	1689
5.19.275	chart.series.i[].datasetIndex.....	1690

5.19.276	chart.series.i[].detail.backgroundColor.....	1690
5.19.277	chart.series.i[].detail.borderColor .....	1691
5.19.278	chart.series.i[].detail.borderRadius .....	1691
5.19.279	chart.series.i[].detail.borderWidth.....	1691
5.19.280	chart.series.i[].detail.color .....	1692
5.19.281	chart.series.i[].detail.fontFamily.....	1692
5.19.282	chart.series.i[].detail.fontSize .....	1692
5.19.283	chart.series.i[].detail.fontStyle.....	1693
	Erlaubte Werte .....	1693
5.19.284	chart.series.i[].detail.fontWeight .....	1693
	Erlaubte Werte .....	1694
5.19.285	chart.series.i[].detail.height .....	1694
5.19.286	chart.series.i[].detail.lineHeight.....	1694
5.19.287	chart.series.i[].detail.object .....	1695
5.19.288	chart.series.i[].detail.offsetCenter .....	1695
	Erlaubte Werte .....	1696
5.19.289	chart.series.i[].detail.padding.....	1696
5.19.290	chart.series.i[].detail.rich.object .....	1696
5.19.291	chart.series.i[].detail.shadowBlur .....	1697
5.19.292	chart.series.i[].detail.shadowColor.....	1698
5.19.293	chart.series.i[].detail.shadowOffsetX.....	1698
5.19.294	chart.series.i[].detail.shadowOffsetY.....	1698
5.19.295	chart.series.i[].detail.show .....	1699
5.19.296	chart.series.i[].detail.textBorderColor .....	1699
5.19.297	chart.series.i[].detail.textBorderWidth .....	1699
5.19.298	chart.series.i[].detail.textShadowBlur .....	1700
5.19.299	chart.series.i[].detail.textShadowColor.....	1700
5.19.300	chart.series.i[].detail.textShadowOffsetX.....	1700
5.19.301	chart.series.i[].detail.textShadowOffsetY.....	1701
5.19.302	chart.series.i[].detail.width .....	1701
5.19.303	chart.series.i[].dimensions .....	1702
5.19.304	chart.series.i[].downplay.itemStyle.borderColor .....	1703
5.19.305	chart.series.i[].downplay.itemStyle.borderType .....	1704
	Erlaubte Werte .....	1704
5.19.306	chart.series.i[].downplay.itemStyle.borderWidth.....	1704
5.19.307	chart.series.i[].downplay.itemStyle.color .....	1704

5.19.308	chart.series.i[].downplay.itemStyle.object .....	1706
5.19.309	chart.series.i[].downplay.itemStyle.opacity .....	1706
5.19.310	chart.series.i[].downplay.itemStyle.shadowBlur .....	1706
5.19.311	chart.series.i[].downplay.itemStyle.shadowColor .....	1707
5.19.312	chart.series.i[].downplay.itemStyle.shadowOffsetX .....	1707
5.19.313	chart.series.i[].downplay.itemStyle.shadowOffsetY .....	1708
5.19.314	chart.series.i[].downplay.label.align .....	1708
	Erlaubte Werte .....	1708
5.19.315	chart.series.i[].downplay.label.backgroundColor .....	1708
5.19.316	chart.series.i[].downplay.label.borderColor .....	1709
5.19.317	chart.series.i[].downplay.label.borderRadius .....	1710
5.19.318	chart.series.i[].downplay.label.borderWidth .....	1710
5.19.319	chart.series.i[].downplay.label.color .....	1710
5.19.320	chart.series.i[].downplay.label.distance .....	1711
5.19.321	chart.series.i[].downplay.label.fontFamily .....	1711
5.19.322	chart.series.i[].downplay.label.fontSize .....	1711
5.19.323	chart.series.i[].downplay.label.fontStyle .....	1712
	Erlaubte Werte .....	1712
5.19.324	chart.series.i[].downplay.label.fontWeight .....	1712
	Erlaubte Werte .....	1713
5.19.325	chart.series.i[].downplay.label.formatter .....	1713
5.19.326	chart.series.i[].downplay.label.height .....	1714
5.19.327	chart.series.i[].downplay.label.lineHeight .....	1714
5.19.328	chart.series.i[].downplay.label.minAngle .....	1715
5.19.329	chart.series.i[].downplay.label.object .....	1715
5.19.330	chart.series.i[].downplay.label.offset .....	1716
5.19.331	chart.series.i[].downplay.label.padding .....	1716
5.19.332	chart.series.i[].downplay.label.position .....	1717
	Erlaubte Werte .....	1717
5.19.333	chart.series.i[].downplay.label.rich.object .....	1717
5.19.334	chart.series.i[].downplay.label.rotate .....	1718
	Erlaubte Werte .....	1719
5.19.335	chart.series.i[].downplay.label.shadowBlur .....	1719
5.19.336	chart.series.i[].downplay.label.shadowColor .....	1719
5.19.337	chart.series.i[].downplay.label.shadowOffsetX .....	1720
5.19.338	chart.series.i[].downplay.label.shadowOffsetY .....	1720

5.19.339	chart.series.i[].downplay.label.show .....	1720
5.19.340	chart.series.i[].downplay.label.textBorderColor .....	1721
5.19.341	chart.series.i[].downplay.label.textBorderWidth .....	1721
5.19.342	chart.series.i[].downplay.label.textShadowBlur .....	1722
5.19.343	chart.series.i[].downplay.label.textShadowColor .....	1722
5.19.344	chart.series.i[].downplay.label.textShadowOffsetX .....	1722
5.19.345	chart.series.i[].downplay.label.textShadowOffsetY .....	1723
5.19.346	chart.series.i[].downplay.label.verticalAlign .....	1723
	Erlaubte Werte .....	1724
5.19.347	chart.series.i[].downplay.label.width .....	1724
5.19.348	chart.series.i[].downplay.object .....	1724
5.19.349	chart.series.i[].draggable .....	1725
5.19.350	chart.series.i[].drillDownIcon .....	1725
5.19.351	chart.series.i[].edgeLabel.align .....	1725
	Erlaubte Werte .....	1726
5.19.352	chart.series.i[].edgeLabel.backgroundColor .....	1726
5.19.353	chart.series.i[].edgeLabel.borderColor .....	1727
5.19.354	chart.series.i[].edgeLabel.borderRadius .....	1727
5.19.355	chart.series.i[].edgeLabel.borderWidth .....	1727
5.19.356	chart.series.i[].edgeLabel.color .....	1728
5.19.357	chart.series.i[].edgeLabel.fontFamily .....	1728
5.19.358	chart.series.i[].edgeLabel.fontSize .....	1728
5.19.359	chart.series.i[].edgeLabel.fontStyle .....	1729
	Erlaubte Werte .....	1729
5.19.360	chart.series.i[].edgeLabel.fontWeight .....	1729
	Erlaubte Werte .....	1730
5.19.361	chart.series.i[].edgeLabel.formatter .....	1730
5.19.362	chart.series.i[].edgeLabel.height .....	1731
5.19.363	chart.series.i[].edgeLabel.lineHeight .....	1731
5.19.364	chart.series.i[].edgeLabel.object .....	1732
5.19.365	chart.series.i[].edgeLabel.padding .....	1732
5.19.366	chart.series.i[].edgeLabel.position .....	1733
	Erlaubte Werte .....	1733
5.19.367	chart.series.i[].edgeLabel.rich.object .....	1733
5.19.368	chart.series.i[].edgeLabel.shadowBlur .....	1734
5.19.369	chart.series.i[].edgeLabel.shadowColor .....	1735

5.19.370	chart.series.i[].edgeLabel.shadowOffsetX.....	1735
5.19.371	chart.series.i[].edgeLabel.shadowOffsetY.....	1735
5.19.372	chart.series.i[].edgeLabel.show.....	1736
5.19.373	chart.series.i[].edgeLabel.textBorderColor.....	1736
5.19.374	chart.series.i[].edgeLabel.textBorderWidth.....	1736
5.19.375	chart.series.i[].edgeLabel.textShadowBlur.....	1737
5.19.376	chart.series.i[].edgeLabel.textShadowColor.....	1737
5.19.377	chart.series.i[].edgeLabel.textShadowOffsetX.....	1737
5.19.378	chart.series.i[].edgeLabel.textShadowOffsetY.....	1738
5.19.379	chart.series.i[].edgeLabel.verticalAlign.....	1738
	Erlaubte Werte.....	1739
5.19.380	chart.series.i[].edgeLabel.width.....	1739
5.19.381	chart.series.i[].edges.....	1739
5.19.382	chart.series.i[].edgeSymbol.....	1740
	Erlaubte Werte.....	1740
5.19.383	chart.series.i[].edgeSymbolSize.....	1740
5.19.384	chart.series.i[].effect.color.....	1741
5.19.385	chart.series.i[].effect.constantSpeed.....	1741
5.19.386	chart.series.i[].effect.delay.....	1742
5.19.387	chart.series.i[].effect.loop.....	1742
5.19.388	chart.series.i[].effect.object.....	1742
5.19.389	chart.series.i[].effect.period.....	1743
5.19.390	chart.series.i[].effect.show.....	1743
5.19.391	chart.series.i[].effect.symbol.....	1743
	Erlaubte Werte.....	1744
5.19.392	chart.series.i[].effect.symbolSize.....	1744
5.19.393	chart.series.i[].effect.trailLength.....	1745
5.19.394	chart.series.i[].effectType.....	1745
	Erlaubte Werte.....	1745
5.19.395	chart.series.i[].emphasis.edgeLabel.align.....	1745
	Erlaubte Werte.....	1746
5.19.396	chart.series.i[].emphasis.edgeLabel.backgroundColor.....	1746
5.19.397	chart.series.i[].emphasis.edgeLabel.borderColor.....	1747
5.19.398	chart.series.i[].emphasis.edgeLabel.borderRadius.....	1747
5.19.399	chart.series.i[].emphasis.edgeLabel.borderWidth.....	1748
5.19.400	chart.series.i[].emphasis.edgeLabel.color.....	1748

5.19.401	chart.series.i[].emphasis.edgeLabel.fontFamily .....	1748
5.19.402	chart.series.i[].emphasis.edgeLabel.fontSize .....	1749
5.19.403	chart.series.i[].emphasis.edgeLabel.fontStyle .....	1749
	Erlaubte Werte .....	1749
5.19.404	chart.series.i[].emphasis.edgeLabel.fontWeight .....	1749
	Erlaubte Werte .....	1750
5.19.405	chart.series.i[].emphasis.edgeLabel.formatter .....	1750
5.19.406	chart.series.i[].emphasis.edgeLabel.height .....	1751
5.19.407	chart.series.i[].emphasis.edgeLabel.lineHeight .....	1752
5.19.408	chart.series.i[].emphasis.edgeLabel.object .....	1752
5.19.409	chart.series.i[].emphasis.edgeLabel.padding .....	1753
5.19.410	chart.series.i[].emphasis.edgeLabel.position .....	1753
	Erlaubte Werte .....	1753
5.19.411	chart.series.i[].emphasis.edgeLabel.rich.object .....	1753
5.19.412	chart.series.i[].emphasis.edgeLabel.shadowBlur .....	1754
5.19.413	chart.series.i[].emphasis.edgeLabel.shadowColor .....	1755
5.19.414	chart.series.i[].emphasis.edgeLabel.shadowOffsetX .....	1755
5.19.415	chart.series.i[].emphasis.edgeLabel.shadowOffsetY .....	1756
5.19.416	chart.series.i[].emphasis.edgeLabel.show .....	1756
5.19.417	chart.series.i[].emphasis.edgeLabel.textBorderColor .....	1756
5.19.418	chart.series.i[].emphasis.edgeLabel.textBorderWidth .....	1757
5.19.419	chart.series.i[].emphasis.edgeLabel.textShadowBlur .....	1757
5.19.420	chart.series.i[].emphasis.edgeLabel.textShadowColor .....	1757
5.19.421	chart.series.i[].emphasis.edgeLabel.textShadowOffsetX .....	1758
5.19.422	chart.series.i[].emphasis.edgeLabel.textShadowOffsetY .....	1758
5.19.423	chart.series.i[].emphasis.edgeLabel.verticalAlign .....	1758
	Erlaubte Werte .....	1759
5.19.424	chart.series.i[].emphasis.edgeLabel.width .....	1759
5.19.425	chart.series.i[].emphasis.itemStyle.borderColor .....	1760
5.19.426	chart.series.i[].emphasis.itemStyle.borderType .....	1760
	Erlaubte Werte .....	1760
5.19.427	chart.series.i[].emphasis.itemStyle.borderWidth .....	1760
5.19.428	chart.series.i[].emphasis.itemStyle.color .....	1761
5.19.429	chart.series.i[].emphasis.itemStyle.object .....	1762
5.19.430	chart.series.i[].emphasis.itemStyle.opacity .....	1763
5.19.431	chart.series.i[].emphasis.itemStyle.shadowBlur .....	1763



5.19.432	chart.series.i[].emphasis.itemStyle.shadowColor .....	1764
5.19.433	chart.series.i[].emphasis.itemStyle.shadowOffsetX .....	1764
5.19.434	chart.series.i[].emphasis.itemStyle.shadowOffsetY .....	1764
5.19.435	chart.series.i[].emphasis.label.align .....	1765
	Erlaubte Werte .....	1765
5.19.436	chart.series.i[].emphasis.label.backgroundColor .....	1765
5.19.437	chart.series.i[].emphasis.label.borderColor .....	1766
5.19.438	chart.series.i[].emphasis.label.borderRadius .....	1766
5.19.439	chart.series.i[].emphasis.label.borderWidth .....	1767
5.19.440	chart.series.i[].emphasis.label.color .....	1767
5.19.441	chart.series.i[].emphasis.label.distance .....	1767
5.19.442	chart.series.i[].emphasis.label.fontFamily .....	1768
5.19.443	chart.series.i[].emphasis.label.fontSize .....	1768
5.19.444	chart.series.i[].emphasis.label.fontStyle .....	1769
	Erlaubte Werte .....	1769
5.19.445	chart.series.i[].emphasis.label.fontWeight .....	1769
	Erlaubte Werte .....	1769
5.19.446	chart.series.i[].emphasis.label.formatter .....	1770
5.19.447	chart.series.i[].emphasis.label.height .....	1771
5.19.448	chart.series.i[].emphasis.label.lineHeight .....	1771
5.19.449	chart.series.i[].emphasis.label.minAngle .....	1772
5.19.450	chart.series.i[].emphasis.label.object .....	1772
5.19.451	chart.series.i[].emphasis.label.offset .....	1773
5.19.452	chart.series.i[].emphasis.label.padding .....	1773
5.19.453	chart.series.i[].emphasis.label.position .....	1773
	Erlaubte Werte .....	1774
5.19.454	chart.series.i[].emphasis.label.rich.object .....	1774
5.19.455	chart.series.i[].emphasis.label.rotate .....	1775
5.19.456	chart.series.i[].emphasis.label.shadowBlur .....	1776
5.19.457	chart.series.i[].emphasis.label.shadowColor .....	1776
5.19.458	chart.series.i[].emphasis.label.shadowOffsetX .....	1776
5.19.459	chart.series.i[].emphasis.label.shadowOffsetY .....	1777
5.19.460	chart.series.i[].emphasis.label.show .....	1777
5.19.461	chart.series.i[].emphasis.label.textBorderColor .....	1777
5.19.462	chart.series.i[].emphasis.label.textBorderWidth .....	1778
5.19.463	chart.series.i[].emphasis.label.textShadowBlur .....	1778

5.19.464	chart.series.i[].emphasis.label.textShadowColor .....	1779
5.19.465	chart.series.i[].emphasis.label.textShadowOffsetX .....	1779
5.19.466	chart.series.i[].emphasis.label.textShadowOffsetY .....	1779
5.19.467	chart.series.i[].emphasis.label.verticalAlign .....	1780
	Erlaubte Werte .....	1780
5.19.468	chart.series.i[].emphasis.label.width .....	1780
5.19.469	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.color .....	1781
5.19.470	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.object .....	1782
5.19.471	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.opacity .....	1783
5.19.472	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowBlur .....	1783
5.19.473	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowColor .....	1784
5.19.474	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetX .....	1784
5.19.475	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetY .....	1784
5.19.476	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.type .....	1785
	Erlaubte Werte .....	1785
5.19.477	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.width .....	1785
5.19.478	chart.series.i[].emphasis.labelLine.emphasis.object .....	1785
5.19.479	chart.series.i[].emphasis.labelLine.emphasis.show .....	1786
5.19.480	chart.series.i[].emphasis.labelLine.length .....	1786
5.19.481	chart.series.i[].emphasis.labelLine.lineStyle.color .....	1786
5.19.482	chart.series.i[].emphasis.labelLine.lineStyle.object .....	1788
5.19.483	chart.series.i[].emphasis.labelLine.lineStyle.opacity .....	1789
5.19.484	chart.series.i[].emphasis.labelLine.lineStyle.shadowBlur .....	1789
5.19.485	chart.series.i[].emphasis.labelLine.lineStyle.shadowColor .....	1790
5.19.486	chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetX .....	1790
5.19.487	chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetY .....	1790
5.19.488	chart.series.i[].emphasis.labelLine.lineStyle.type .....	1791
	Erlaubte Werte .....	1791
5.19.489	chart.series.i[].emphasis.labelLine.lineStyle.width .....	1791
5.19.490	chart.series.i[].emphasis.labelLine.object .....	1791
5.19.491	chart.series.i[].emphasis.labelLine.show .....	1792
5.19.492	chart.series.i[].emphasis.lineStyle.color .....	1792
5.19.493	chart.series.i[].emphasis.lineStyle.curveness .....	1793
5.19.494	chart.series.i[].emphasis.lineStyle.object .....	1794
5.19.495	chart.series.i[].emphasis.lineStyle.shadowBlur .....	1794
5.19.496	chart.series.i[].emphasis.lineStyle.shadowColor .....	1795

5.19.497	chart.series.i[].emphasis.lineStyle.shadowOffsetX.....	1795
5.19.498	chart.series.i[].emphasis.lineStyle.shadowOffsetY.....	1795
5.19.499	chart.series.i[].emphasis.lineStyle.width.....	1796
5.19.500	chart.series.i[].emphasis.object.....	1796
5.19.501	chart.series.i[].emphasis.upperLabel.align.....	1796
	Erlaubte Werte.....	1797
5.19.502	chart.series.i[].emphasis.upperLabel.backgroundColor.....	1797
5.19.503	chart.series.i[].emphasis.upperLabel.borderColor.....	1798
5.19.504	chart.series.i[].emphasis.upperLabel.borderRadius.....	1798
5.19.505	chart.series.i[].emphasis.upperLabel.borderWidth.....	1798
5.19.506	chart.series.i[].emphasis.upperLabel.color.....	1799
5.19.507	chart.series.i[].emphasis.upperLabel.distance.....	1799
5.19.508	chart.series.i[].emphasis.upperLabel.ellipsis.....	1800
5.19.509	chart.series.i[].emphasis.upperLabel.fontFamily.....	1800
5.19.510	chart.series.i[].emphasis.upperLabel.fontSize.....	1800
5.19.511	chart.series.i[].emphasis.upperLabel.fontStyle.....	1801
	Erlaubte Werte.....	1801
5.19.512	chart.series.i[].emphasis.upperLabel.fontWeight.....	1801
	Erlaubte Werte.....	1802
5.19.513	chart.series.i[].emphasis.upperLabel.formatter.....	1802
5.19.514	chart.series.i[].emphasis.upperLabel.height.....	1803
5.19.515	chart.series.i[].emphasis.upperLabel.lineHeight.....	1803
5.19.516	chart.series.i[].emphasis.upperLabel.object.....	1804
5.19.517	chart.series.i[].emphasis.upperLabel.offset.....	1804
5.19.518	chart.series.i[].emphasis.upperLabel.padding.....	1805
5.19.519	chart.series.i[].emphasis.upperLabel.position.....	1805
	Erlaubte Werte.....	1806
5.19.520	chart.series.i[].emphasis.upperLabel.rich.object.....	1806
5.19.521	chart.series.i[].emphasis.upperLabel.rotate.....	1807
5.19.522	chart.series.i[].emphasis.upperLabel.shadowBlur.....	1808
5.19.523	chart.series.i[].emphasis.upperLabel.shadowColor.....	1808
5.19.524	chart.series.i[].emphasis.upperLabel.shadowOffsetX.....	1808
5.19.525	chart.series.i[].emphasis.upperLabel.shadowOffsetY.....	1809
5.19.526	chart.series.i[].emphasis.upperLabel.show.....	1809
5.19.527	chart.series.i[].emphasis.upperLabel.textBorderColor.....	1809
5.19.528	chart.series.i[].emphasis.upperLabel.textBorderWidth.....	1810

5.19.529	chart.series.i[].emphasis.upperLabel.textShadowBlur .....	1810
5.19.530	chart.series.i[].emphasis.upperLabel.textShadowColor .....	1810
5.19.531	chart.series.i[].emphasis.upperLabel.textShadowOffsetX .....	1811
5.19.532	chart.series.i[].emphasis.upperLabel.textShadowOffsetY .....	1811
5.19.533	chart.series.i[].emphasis.upperLabel.verticalAlign .....	1812
	Erlaubte Werte .....	1812
5.19.534	chart.series.i[].emphasis.upperLabel.width.....	1812
5.19.535	chart.series.i[].encode .....	1813
5.19.536	chart.series.i[].endAngle.....	1815
5.19.537	chart.series.i[].expandAndCollapse .....	1815
5.19.538	chart.series.i[].focusNodeAdjacency .....	1815
	Erlaubte Werte .....	1816
5.19.539	chart.series.i[].force.edgeLength .....	1816
5.19.540	chart.series.i[].force.gravity .....	1817
5.19.541	chart.series.i[].force.initLayout.....	1817
5.19.542	chart.series.i[].force.layoutAnimation .....	1817
5.19.543	chart.series.i[].force.object .....	1818
5.19.544	chart.series.i[].force.repulsion .....	1818
5.19.545	chart.series.i[].funnelAlign .....	1819
	Erlaubte Werte .....	1819
5.19.546	chart.series.i[].gap .....	1819
5.19.547	chart.series.i[].geoIndex .....	1819
5.19.548	chart.series.i[].height.....	1820
5.19.549	chart.series.i[].highlight.itemStyle.borderColor .....	1820
5.19.550	chart.series.i[].highlight.itemStyle.borderType.....	1821
	Erlaubte Werte .....	1821
5.19.551	chart.series.i[].highlight.itemStyle.borderWidth .....	1821
5.19.552	chart.series.i[].highlight.itemStyle.color .....	1821
5.19.553	chart.series.i[].highlight.itemStyle.object .....	1823
5.19.554	chart.series.i[].highlight.itemStyle.opacity .....	1823
5.19.555	chart.series.i[].highlight.itemStyle.shadowBlur .....	1823
5.19.556	chart.series.i[].highlight.itemStyle.shadowColor .....	1824
5.19.557	chart.series.i[].highlight.itemStyle.shadowOffsetX .....	1824
5.19.558	chart.series.i[].highlight.itemStyle.shadowOffsetY .....	1825
5.19.559	chart.series.i[].highlight.label.align .....	1825
	Erlaubte Werte .....	1825

5.19.560	chart.series.i[].highlight.label.backgroundColor .....	1825
5.19.561	chart.series.i[].highlight.label.borderColor .....	1826
5.19.562	chart.series.i[].highlight.label.borderRadius.....	1827
5.19.563	chart.series.i[].highlight.label.borderWidth .....	1827
5.19.564	chart.series.i[].highlight.label.color .....	1827
5.19.565	chart.series.i[].highlight.label.distance .....	1828
5.19.566	chart.series.i[].highlight.label.fontFamily .....	1828
5.19.567	chart.series.i[].highlight.label.fontSize.....	1828
5.19.568	chart.series.i[].highlight.label.fontStyle .....	1829
	Erlaubte Werte .....	1829
5.19.569	chart.series.i[].highlight.label.fontWeight.....	1829
	Erlaubte Werte .....	1830
5.19.570	chart.series.i[].highlight.label.formatter .....	1830
5.19.571	chart.series.i[].highlight.label.height.....	1831
5.19.572	chart.series.i[].highlight.label.lineHeight .....	1831
5.19.573	chart.series.i[].highlight.label.minAngle .....	1832
5.19.574	chart.series.i[].highlight.label.object.....	1832
5.19.575	chart.series.i[].highlight.label.offset.....	1833
5.19.576	chart.series.i[].highlight.label.padding .....	1833
5.19.577	chart.series.i[].highlight.label.position.....	1834
	Erlaubte Werte .....	1834
5.19.578	chart.series.i[].highlight.label.rich.object .....	1834
5.19.579	chart.series.i[].highlight.label.rotate .....	1835
	Erlaubte Werte .....	1836
5.19.580	chart.series.i[].highlight.label.shadowBlur .....	1836
5.19.581	chart.series.i[].highlight.label.shadowColor .....	1836
5.19.582	chart.series.i[].highlight.label.shadowOffsetX .....	1837
5.19.583	chart.series.i[].highlight.label.shadowOffsetY .....	1837
5.19.584	chart.series.i[].highlight.label.show .....	1837
5.19.585	chart.series.i[].highlight.label.textBorderColor .....	1838
5.19.586	chart.series.i[].highlight.label.textBorderWidth .....	1838
5.19.587	chart.series.i[].highlight.label.textShadowBlur .....	1839
5.19.588	chart.series.i[].highlight.label.textShadowColor .....	1839
5.19.589	chart.series.i[].highlight.label.textShadowOffsetX .....	1839
5.19.590	chart.series.i[].highlight.label.textShadowOffsetY .....	1840
5.19.591	chart.series.i[].highlight.label.verticalAlign .....	1840

Erlaubte Werte .....	1841
5.19.592 chart.series.i[].highlight.label.width.....	1841
5.19.593 chart.series.i[].highlight.object.....	1841
5.19.594 chart.series.i[].highlightPolicy .....	1842
Erlaubte Werte .....	1842
5.19.595 chart.series.i[].hoverAnimation .....	1842
5.19.596 chart.series.i[].hoverAnimation.animation .....	1843
5.19.597 chart.series.i[].hoverAnimation.animationDelay.....	1843
5.19.598 chart.series.i[].hoverAnimation.animationDelayUpdate .....	1844
5.19.599 chart.series.i[].hoverAnimation.animationDuration .....	1844
5.19.600 chart.series.i[].hoverAnimation.animationDurationUpdate .....	1845
5.19.601 chart.series.i[].hoverAnimation.animationEasing .....	1845
Erlaubte Werte .....	1846
5.19.602 chart.series.i[].hoverAnimation.animationEasingUpdate.....	1846
Erlaubte Werte .....	1847
5.19.603 chart.series.i[].hoverAnimation.animationThreshold .....	1847
5.19.604 chart.series.i[].hoverAnimation.object.....	1848
5.19.605 chart.series.i[].hoverOffset.....	1848
5.19.606 chart.series.i[].id .....	1849
5.19.607 chart.series.i[].inactiveOpacity .....	1849
5.19.608 chart.series.i[].initialTreeDepth .....	1850
5.19.609 chart.series.i[].itemStyle.borderColor .....	1850
5.19.610 chart.series.i[].itemStyle.borderType.....	1850
Erlaubte Werte .....	1851
5.19.611 chart.series.i[].itemStyle.borderWidth .....	1851
5.19.612 chart.series.i[].itemStyle.color .....	1851
5.19.613 chart.series.i[].itemStyle.object.....	1852
5.19.614 chart.series.i[].itemStyle.opacity .....	1853
5.19.615 chart.series.i[].itemStyle.shadowBlur .....	1853
5.19.616 chart.series.i[].itemStyle.shadowColor .....	1854
5.19.617 chart.series.i[].itemStyle.shadowOffsetX .....	1854
5.19.618 chart.series.i[].itemStyle.shadowOffsetY .....	1854
5.19.619 chart.series.i[].label.align .....	1855
Erlaubte Werte .....	1855
5.19.620 chart.series.i[].label.backgroundColor .....	1855
5.19.621 chart.series.i[].label.borderColor.....	1856

5.19.622	chart.series.i[].label.borderRadius .....	1856
5.19.623	chart.series.i[].label.borderWidth .....	1857
5.19.624	chart.series.i[].label.color .....	1857
5.19.625	chart.series.i[].label.distance .....	1857
5.19.626	chart.series.i[].label.fontFamily .....	1858
5.19.627	chart.series.i[].label.fontSize .....	1858
5.19.628	chart.series.i[].label.fontStyle .....	1859
	Erlaubte Werte .....	1859
5.19.629	chart.series.i[].label.fontWeight .....	1859
	Erlaubte Werte .....	1859
5.19.630	chart.series.i[].label.formatter .....	1860
5.19.631	chart.series.i[].label.height .....	1861
5.19.632	chart.series.i[].label.lineHeight .....	1861
5.19.633	chart.series.i[].label.minAngle .....	1862
5.19.634	chart.series.i[].label.object .....	1862
5.19.635	chart.series.i[].label.offset .....	1863
5.19.636	chart.series.i[].label.padding .....	1863
5.19.637	chart.series.i[].label.position .....	1864
	Erlaubte Werte .....	1864
5.19.638	chart.series.i[].label.rich.object .....	1864
5.19.639	chart.series.i[].label.rotate .....	1865
5.19.640	chart.series.i[].label.shadowBlur .....	1866
5.19.641	chart.series.i[].label.shadowColor .....	1866
5.19.642	chart.series.i[].label.shadowOffsetX .....	1866
5.19.643	chart.series.i[].label.shadowOffsetY .....	1867
5.19.644	chart.series.i[].label.show .....	1867
5.19.645	chart.series.i[].label.textBorderColor .....	1867
5.19.646	chart.series.i[].label.textBorderWidth .....	1868
5.19.647	chart.series.i[].label.textShadowBlur .....	1868
5.19.648	chart.series.i[].label.textShadowColor .....	1869
5.19.649	chart.series.i[].label.textShadowOffsetX .....	1869
5.19.650	chart.series.i[].label.textShadowOffsetY .....	1869
5.19.651	chart.series.i[].label.verticalAlign .....	1870
	Erlaubte Werte .....	1870
5.19.652	chart.series.i[].label.width .....	1870
5.19.653	chart.series.i[].labelLine.emphasis.lineStyle.color .....	1871

5.19.654	chart.series.i[].labelLine.emphasis.lineStyle.object .....	1872
5.19.655	chart.series.i[].labelLine.emphasis.lineStyle.opacity .....	1873
5.19.656	chart.series.i[].labelLine.emphasis.lineStyle.shadowBlur .....	1873
5.19.657	chart.series.i[].labelLine.emphasis.lineStyle.shadowColor .....	1874
5.19.658	chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetX .....	1874
5.19.659	chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetY .....	1874
5.19.660	chart.series.i[].labelLine.emphasis.lineStyle.type .....	1875
	Erlaubte Werte .....	1875
5.19.661	chart.series.i[].labelLine.emphasis.lineStyle.width .....	1875
5.19.662	chart.series.i[].labelLine.emphasis.object .....	1875
5.19.663	chart.series.i[].labelLine.emphasis.show .....	1876
5.19.664	chart.series.i[].labelLine.length .....	1876
5.19.665	chart.series.i[].labelLine.length2 .....	1876
5.19.666	chart.series.i[].labelLine.lineStyle.color .....	1877
5.19.667	chart.series.i[].labelLine.lineStyle.object .....	1878
5.19.668	chart.series.i[].labelLine.lineStyle.opacity .....	1879
5.19.669	chart.series.i[].labelLine.lineStyle.shadowBlur .....	1879
5.19.670	chart.series.i[].labelLine.lineStyle.shadowColor .....	1880
5.19.671	chart.series.i[].labelLine.lineStyle.shadowOffsetX .....	1880
5.19.672	chart.series.i[].labelLine.lineStyle.shadowOffsetY .....	1880
5.19.673	chart.series.i[].labelLine.lineStyle.type .....	1881
	Erlaubte Werte .....	1881
5.19.674	chart.series.i[].labelLine.lineStyle.width .....	1881
5.19.675	chart.series.i[].labelLine.object .....	1881
5.19.676	chart.series.i[].labelLine.show .....	1882
5.19.677	chart.series.i[].labelLine.smooth .....	1882
5.19.678	chart.series.i[].large .....	1882
5.19.679	chart.series.i[].largeThreshold .....	1883
5.19.680	chart.series.i[].layout .....	1883
5.19.681	chart.series.i[].layoutIterations .....	1884
5.19.682	chart.series.i[].leafDepth .....	1884
5.19.683	chart.series.i[].leaves.emphasis.itemStyle.borderColor .....	1885
5.19.684	chart.series.i[].leaves.emphasis.itemStyle.borderType .....	1885
	Erlaubte Werte .....	1886
5.19.685	chart.series.i[].leaves.emphasis.itemStyle.borderWidth .....	1886
5.19.686	chart.series.i[].leaves.emphasis.itemStyle.color .....	1886



5.19.687	chart.series.i[].leaves.emphasis.itemStyle.object .....	1887
5.19.688	chart.series.i[].leaves.emphasis.itemStyle.opacity .....	1888
5.19.689	chart.series.i[].leaves.emphasis.itemStyle.shadowBlur .....	1888
5.19.690	chart.series.i[].leaves.emphasis.itemStyle.shadowColor .....	1889
5.19.691	chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetX .....	1889
5.19.692	chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetY .....	1889
5.19.693	chart.series.i[].leaves.emphasis.label.align .....	1890
	Erlaubte Werte .....	1890
5.19.694	chart.series.i[].leaves.emphasis.label.backgroundColor .....	1890
5.19.695	chart.series.i[].leaves.emphasis.label.borderColor .....	1891
5.19.696	chart.series.i[].leaves.emphasis.label.borderRadius .....	1891
5.19.697	chart.series.i[].leaves.emphasis.label.borderWidth .....	1892
5.19.698	chart.series.i[].leaves.emphasis.label.color .....	1892
5.19.699	chart.series.i[].leaves.emphasis.label.distance .....	1892
5.19.700	chart.series.i[].leaves.emphasis.label.fontFamily .....	1893
5.19.701	chart.series.i[].leaves.emphasis.label.fontSize .....	1893
5.19.702	chart.series.i[].leaves.emphasis.label.fontStyle .....	1894
	Erlaubte Werte .....	1894
5.19.703	chart.series.i[].leaves.emphasis.label.fontWeight .....	1894
	Erlaubte Werte .....	1894
5.19.704	chart.series.i[].leaves.emphasis.label.height .....	1895
5.19.705	chart.series.i[].leaves.emphasis.label.lineHeight .....	1895
5.19.706	chart.series.i[].leaves.emphasis.label.object .....	1896
5.19.707	chart.series.i[].leaves.emphasis.label.offset .....	1896
5.19.708	chart.series.i[].leaves.emphasis.label.padding .....	1896
5.19.709	chart.series.i[].leaves.emphasis.label.position .....	1897
	Erlaubte Werte .....	1897
5.19.710	chart.series.i[].leaves.emphasis.label.rich.object .....	1898
5.19.711	chart.series.i[].leaves.emphasis.label.rotate .....	1899
5.19.712	chart.series.i[].leaves.emphasis.label.shadowBlur .....	1900
5.19.713	chart.series.i[].leaves.emphasis.label.shadowColor .....	1900
5.19.714	chart.series.i[].leaves.emphasis.label.shadowOffsetX .....	1900
5.19.715	chart.series.i[].leaves.emphasis.label.shadowOffsetY .....	1901
5.19.716	chart.series.i[].leaves.emphasis.label.show .....	1901
5.19.717	chart.series.i[].leaves.emphasis.label.textBorderColor .....	1901
5.19.718	chart.series.i[].leaves.emphasis.label.textBorderWidth .....	1902

5.19.719	chart.series.i[].leaves.emphasis.label.textShadowBlur .....	1902
5.19.720	chart.series.i[].leaves.emphasis.label.textShadowColor .....	1902
5.19.721	chart.series.i[].leaves.emphasis.label.textShadowOffsetX .....	1903
5.19.722	chart.series.i[].leaves.emphasis.label.textShadowOffsetY .....	1903
5.19.723	chart.series.i[].leaves.emphasis.label.verticalAlign.....	1904
	Erlaubte Werte .....	1904
5.19.724	chart.series.i[].leaves.emphasis.label.width .....	1904
5.19.725	chart.series.i[].leaves.emphasis.object .....	1905
5.19.726	chart.series.i[].leaves.itemStyle.borderColor .....	1905
5.19.727	chart.series.i[].leaves.itemStyle.borderType .....	1905
	Erlaubte Werte .....	1906
5.19.728	chart.series.i[].leaves.itemStyle.borderWidth.....	1906
5.19.729	chart.series.i[].leaves.itemStyle.color .....	1906
5.19.730	chart.series.i[].leaves.itemStyle.object .....	1908
5.19.731	chart.series.i[].leaves.itemStyle.opacity .....	1908
5.19.732	chart.series.i[].leaves.itemStyle.shadowBlur.....	1908
5.19.733	chart.series.i[].leaves.itemStyle.shadowColor.....	1909
5.19.734	chart.series.i[].leaves.itemStyle.shadowOffsetX.....	1909
5.19.735	chart.series.i[].leaves.itemStyle.shadowOffsetY.....	1909
5.19.736	chart.series.i[].leaves.label.align .....	1910
	Erlaubte Werte .....	1910
5.19.737	chart.series.i[].leaves.label.backgroundColor .....	1910
5.19.738	chart.series.i[].leaves.label.borderColor .....	1911
5.19.739	chart.series.i[].leaves.label.borderRadius.....	1912
5.19.740	chart.series.i[].leaves.label.borderWidth .....	1912
5.19.741	chart.series.i[].leaves.label.color .....	1912
5.19.742	chart.series.i[].leaves.label.distance .....	1913
5.19.743	chart.series.i[].leaves.label.fontFamily.....	1913
5.19.744	chart.series.i[].leaves.label.fontSize .....	1913
5.19.745	chart.series.i[].leaves.label.fontStyle .....	1914
	Erlaubte Werte .....	1914
5.19.746	chart.series.i[].leaves.label.fontWeight .....	1914
	Erlaubte Werte .....	1915
5.19.747	chart.series.i[].leaves.label.height .....	1915
5.19.748	chart.series.i[].leaves.label.lineHeight .....	1915
5.19.749	chart.series.i[].leaves.label.object .....	1916

5.19.750	chart.series.i[].leaves.label.offset .....	1916
5.19.751	chart.series.i[].leaves.label.padding .....	1916
5.19.752	chart.series.i[].leaves.label.position .....	1917
	Erlaubte Werte .....	1917
5.19.753	chart.series.i[].leaves.label.rich.object .....	1918
5.19.754	chart.series.i[].leaves.label.rotate .....	1919
5.19.755	chart.series.i[].leaves.label.shadowBlur .....	1920
5.19.756	chart.series.i[].leaves.label.shadowColor .....	1920
5.19.757	chart.series.i[].leaves.label.shadowOffsetX .....	1920
5.19.758	chart.series.i[].leaves.label.shadowOffsetY .....	1921
5.19.759	chart.series.i[].leaves.label.show .....	1921
5.19.760	chart.series.i[].leaves.label.textBorderColor .....	1921
5.19.761	chart.series.i[].leaves.label.textBorderWidth .....	1922
5.19.762	chart.series.i[].leaves.label.textShadowBlur .....	1922
5.19.763	chart.series.i[].leaves.label.textShadowColor .....	1922
5.19.764	chart.series.i[].leaves.label.textShadowOffsetX .....	1923
5.19.765	chart.series.i[].leaves.label.textShadowOffsetY .....	1923
5.19.766	chart.series.i[].leaves.label.verticalAlign .....	1924
	Erlaubte Werte .....	1924
5.19.767	chart.series.i[].leaves.label.width .....	1924
5.19.768	chart.series.i[].leaves.object .....	1925
5.19.769	chart.series.i[].left .....	1925
	Erlaubte Werte .....	1926
5.19.770	chart.series.i[].legendHoverLink .....	1926
5.19.771	chart.series.i[].levels.childrenVisibleMin .....	1926
5.19.772	chart.series.i[].levels.colorAlpha .....	1927
5.19.773	chart.series.i[].levels.colorMappingBy .....	1927
	Erlaubte Werte .....	1928
5.19.774	chart.series.i[].levels.colorSaturation .....	1928
5.19.775	chart.series.i[].levels.emphasis.itemStyle.color .....	1929
5.19.776	chart.series.i[].levels.emphasis.itemStyle.object .....	1929
5.19.777	chart.series.i[].levels.emphasis.label.align .....	1929
	Erlaubte Werte .....	1930
5.19.778	chart.series.i[].levels.emphasis.label.backgroundColor .....	1930
5.19.779	chart.series.i[].levels.emphasis.label.borderColor .....	1931
5.19.780	chart.series.i[].levels.emphasis.label.borderRadius .....	1931

5.19.781	chart.series.i[].levels.emphasis.label.borderWidth .....	1932
5.19.782	chart.series.i[].levels.emphasis.label.color .....	1932
5.19.783	chart.series.i[].levels.emphasis.label.distance .....	1932
5.19.784	chart.series.i[].levels.emphasis.label.ellipsis .....	1933
5.19.785	chart.series.i[].levels.emphasis.label.fontFamily .....	1933
5.19.786	chart.series.i[].levels.emphasis.label.fontSize .....	1933
5.19.787	chart.series.i[].levels.emphasis.label.fontStyle .....	1934
	Erlaubte Werte .....	1934
5.19.788	chart.series.i[].levels.emphasis.label.fontWeight .....	1934
	Erlaubte Werte .....	1935
5.19.789	chart.series.i[].levels.emphasis.label.formatter .....	1935
5.19.790	chart.series.i[].levels.emphasis.label.height .....	1936
5.19.791	chart.series.i[].levels.emphasis.label.lineHeight .....	1937
5.19.792	chart.series.i[].levels.emphasis.label.object .....	1937
5.19.793	chart.series.i[].levels.emphasis.label.offset .....	1937
5.19.794	chart.series.i[].levels.emphasis.label.padding.....	1938
5.19.795	chart.series.i[].levels.emphasis.label.position .....	1938
	Erlaubte Werte .....	1939
5.19.796	chart.series.i[].levels.emphasis.label.rich.object.....	1939
5.19.797	chart.series.i[].levels.emphasis.label.rotate .....	1940
5.19.798	chart.series.i[].levels.emphasis.label.shadowBlur .....	1941
5.19.799	chart.series.i[].levels.emphasis.label.shadowColor .....	1941
5.19.800	chart.series.i[].levels.emphasis.label.shadowOffsetX .....	1941
5.19.801	chart.series.i[].levels.emphasis.label.shadowOffsetY .....	1942
5.19.802	chart.series.i[].levels.emphasis.label.show.....	1942
5.19.803	chart.series.i[].levels.emphasis.label.textBorderColor .....	1942
5.19.804	chart.series.i[].levels.emphasis.label.textBorderWidth.....	1943
5.19.805	chart.series.i[].levels.emphasis.label.textShadowBlur .....	1943
5.19.806	chart.series.i[].levels.emphasis.label.textShadowColor .....	1943
5.19.807	chart.series.i[].levels.emphasis.label.textShadowOffsetX .....	1944
5.19.808	chart.series.i[].levels.emphasis.label.textShadowOffsetY .....	1944
5.19.809	chart.series.i[].levels.emphasis.label.verticalAlign.....	1945
	Erlaubte Werte .....	1945
5.19.810	chart.series.i[].levels.emphasis.label.width .....	1945
5.19.811	chart.series.i[].levels.emphasis.object .....	1946
5.19.812	chart.series.i[].levels.emphasis.upperLabel.align .....	1946

Erlaubte Werte .....	1947
5.19.813 chart.series.i[].levels.emphasis.upperLabel.backgroundColor .....	1947
5.19.814 chart.series.i[].levels.emphasis.upperLabel.borderColor .....	1948
5.19.815 chart.series.i[].levels.emphasis.upperLabel.borderRadius .....	1948
5.19.816 chart.series.i[].levels.emphasis.upperLabel.borderWidth .....	1948
5.19.817 chart.series.i[].levels.emphasis.upperLabel.color .....	1949
5.19.818 chart.series.i[].levels.emphasis.upperLabel.distance .....	1949
5.19.819 chart.series.i[].levels.emphasis.upperLabel.ellipsis .....	1950
5.19.820 chart.series.i[].levels.emphasis.upperLabel.fontFamily .....	1950
5.19.821 chart.series.i[].levels.emphasis.upperLabel.fontSize .....	1950
5.19.822 chart.series.i[].levels.emphasis.upperLabel.fontStyle .....	1951
Erlaubte Werte .....	1951
5.19.823 chart.series.i[].levels.emphasis.upperLabel.fontWeight .....	1951
Erlaubte Werte .....	1952
5.19.824 chart.series.i[].levels.emphasis.upperLabel.formatter .....	1952
5.19.825 chart.series.i[].levels.emphasis.upperLabel.height .....	1953
5.19.826 chart.series.i[].levels.emphasis.upperLabel.lineHeight .....	1953
5.19.827 chart.series.i[].levels.emphasis.upperLabel.object .....	1954
5.19.828 chart.series.i[].levels.emphasis.upperLabel.offset .....	1954
5.19.829 chart.series.i[].levels.emphasis.upperLabel.padding .....	1955
5.19.830 chart.series.i[].levels.emphasis.upperLabel.position .....	1955
Erlaubte Werte .....	1956
5.19.831 chart.series.i[].levels.emphasis.upperLabel.rich.object .....	1956
5.19.832 chart.series.i[].levels.emphasis.upperLabel.rotate .....	1957
5.19.833 chart.series.i[].levels.emphasis.upperLabel.shadowBlur .....	1958
5.19.834 chart.series.i[].levels.emphasis.upperLabel.shadowColor .....	1958
5.19.835 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetX .....	1958
5.19.836 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetY .....	1959
5.19.837 chart.series.i[].levels.emphasis.upperLabel.show .....	1959
5.19.838 chart.series.i[].levels.emphasis.upperLabel.textBorderColor .....	1959
5.19.839 chart.series.i[].levels.emphasis.upperLabel.textBorderWidth .....	1960
5.19.840 chart.series.i[].levels.emphasis.upperLabel.textShadowBlur .....	1960
5.19.841 chart.series.i[].levels.emphasis.upperLabel.textShadowColor .....	1960
5.19.842 chart.series.i[].levels.emphasis.upperLabel.textShadowOffsetX .....	1961
5.19.843 chart.series.i[].levels.emphasis.upperLabel.textShadowOffsetY .....	1961
5.19.844 chart.series.i[].levels.emphasis.upperLabel.verticalAlign .....	1962

Erlaubte Werte .....	1962
5.19.845 chart.series.i[].levels.emphasis.upperLabel.width .....	1962
5.19.846 chart.series.i[].levels.itemStyle.borderColor .....	1963
5.19.847 chart.series.i[].levels.itemStyle.borderColorSaturation.....	1963
5.19.848 chart.series.i[].levels.itemStyle.borderWidth.....	1964
5.19.849 chart.series.i[].levels.itemStyle.color .....	1964
5.19.850 chart.series.i[].levels.itemStyle.colorAlpha.....	1964
5.19.851 chart.series.i[].levels.itemStyle.colorSaturation .....	1965
5.19.852 chart.series.i[].levels.itemStyle.gapWidth.....	1965
5.19.853 chart.series.i[].levels.itemStyle.object .....	1966
5.19.854 chart.series.i[].levels.itemStyle.strokeColor .....	1966
5.19.855 chart.series.i[].levels.itemStyle.strokeWidth .....	1966
5.19.856 chart.series.i[].levels.label.align .....	1967
Erlaubte Werte .....	1967
5.19.857 chart.series.i[].levels.label.backgroundColor .....	1967
5.19.858 chart.series.i[].levels.label.borderColor .....	1968
5.19.859 chart.series.i[].levels.label.borderRadius .....	1968
5.19.860 chart.series.i[].levels.label.borderWidth .....	1969
5.19.861 chart.series.i[].levels.label.color .....	1969
5.19.862 chart.series.i[].levels.label.distance .....	1970
5.19.863 chart.series.i[].levels.label.ellipsis .....	1970
5.19.864 chart.series.i[].levels.label.fontFamily.....	1970
5.19.865 chart.series.i[].levels.label.fontSize .....	1971
5.19.866 chart.series.i[].levels.label.fontStyle .....	1971
5.19.867 chart.series.i[].levels.label.fontWeight .....	1971
Erlaubte Werte .....	1972
5.19.868 chart.series.i[].levels.label.formatter .....	1972
5.19.869 chart.series.i[].levels.label.height.....	1973
5.19.870 chart.series.i[].levels.label.lineHeight .....	1974
5.19.871 chart.series.i[].levels.label.object .....	1974
5.19.872 chart.series.i[].levels.label.offset .....	1975
5.19.873 chart.series.i[].levels.label.padding.....	1975
5.19.874 chart.series.i[].levels.label.position .....	1975
Erlaubte Werte .....	1976
5.19.875 chart.series.i[].levels.label.rich.object.....	1976
5.19.876 chart.series.i[].levels.label.rotate .....	1977

5.19.877	chart.series.i[].levels.label.shadowBlur.....	1978
5.19.878	chart.series.i[].levels.label.shadowColor .....	1978
5.19.879	chart.series.i[].levels.label.shadowOffsetX .....	1978
5.19.880	chart.series.i[].levels.label.shadowOffsetY.....	1979
5.19.881	chart.series.i[].levels.label.show .....	1979
5.19.882	chart.series.i[].levels.label.textBorderColor.....	1979
5.19.883	chart.series.i[].levels.label.textBorderWidth.....	1980
5.19.884	chart.series.i[].levels.label.textShadowBlur.....	1980
5.19.885	chart.series.i[].levels.label.textShadowColor .....	1980
5.19.886	chart.series.i[].levels.label.textShadowOffsetX .....	1981
5.19.887	chart.series.i[].levels.label.textShadowOffsetY.....	1981
5.19.888	chart.series.i[].levels.label.verticalAlign .....	1982
	Erlaubte Werte .....	1982
5.19.889	chart.series.i[].levels.label.width .....	1982
5.19.890	chart.series.i[].levels.object .....	1983
5.19.891	chart.series.i[].levels.upperLabel.align .....	1985
	Erlaubte Werte .....	1986
5.19.892	chart.series.i[].levels.upperLabel.backgroundColor .....	1986
5.19.893	chart.series.i[].levels.upperLabel.borderColor .....	1987
5.19.894	chart.series.i[].levels.upperLabel.borderRadius.....	1987
5.19.895	chart.series.i[].levels.upperLabel.borderWidth .....	1987
5.19.896	chart.series.i[].levels.upperLabel.color .....	1988
5.19.897	chart.series.i[].levels.upperLabel.distance .....	1988
5.19.898	chart.series.i[].levels.upperLabel.ellipsis .....	1989
5.19.899	chart.series.i[].levels.upperLabel.fontFamily.....	1989
5.19.900	chart.series.i[].levels.upperLabel.fontSize .....	1989
5.19.901	chart.series.i[].levels.upperLabel.fontStyle .....	1990
	Erlaubte Werte .....	1990
5.19.902	chart.series.i[].levels.upperLabel.fontWeight .....	1990
	Erlaubte Werte .....	1991
5.19.903	chart.series.i[].levels.upperLabel.formatter.....	1991
5.19.904	chart.series.i[].levels.upperLabel.height .....	1992
5.19.905	chart.series.i[].levels.upperLabel.lineHeight .....	1992
5.19.906	chart.series.i[].levels.upperLabel.object .....	1993
5.19.907	chart.series.i[].levels.upperLabel.offset .....	1993
5.19.908	chart.series.i[].levels.upperLabel.padding.....	1994

5.19.909	chart.series.i[].levels.upperLabel.position .....	1994
	Erlaubte Werte .....	1995
5.19.910	chart.series.i[].levels.upperLabel.rich.object.....	1995
5.19.911	chart.series.i[].levels.upperLabel.rotate .....	1996
5.19.912	chart.series.i[].levels.upperLabel.shadowBlur.....	1997
5.19.913	chart.series.i[].levels.upperLabel.shadowColor .....	1997
5.19.914	chart.series.i[].levels.upperLabel.shadowOffsetX .....	1997
5.19.915	chart.series.i[].levels.upperLabel.shadowOffsetY.....	1998
5.19.916	chart.series.i[].levels.upperLabel.show .....	1998
5.19.917	chart.series.i[].levels.upperLabel.textBorderColor.....	1998
5.19.918	chart.series.i[].levels.upperLabel.textBorderWidth.....	1999
5.19.919	chart.series.i[].levels.upperLabel.textShadowBlur.....	1999
5.19.920	chart.series.i[].levels.upperLabel.textShadowColor.....	1999
5.19.921	chart.series.i[].levels.upperLabel.textShadowOffsetX .....	2000
5.19.922	chart.series.i[].levels.upperLabel.textShadowOffsetY.....	2000
5.19.923	chart.series.i[].levels.upperLabel.verticalAlign.....	2001
	Erlaubte Werte .....	2001
5.19.924	chart.series.i[].levels.upperLabel.width .....	2001
5.19.925	chart.series.i[].levels.visibleMin .....	2002
5.19.926	chart.series.i[].levels.visualDimension .....	2002
5.19.927	chart.series.i[].levels.visualMax .....	2003
5.19.928	chart.series.i[].levels.visualMin .....	2003
5.19.929	chart.series.i[].lineStyle.color .....	2004
5.19.930	chart.series.i[].lineStyle.object .....	2005
5.19.931	chart.series.i[].lineStyle.opacity .....	2006
5.19.932	chart.series.i[].lineStyle.shadowBlur.....	2006
5.19.933	chart.series.i[].lineStyle.shadowColor.....	2007
5.19.934	chart.series.i[].lineStyle.shadowOffsetX.....	2007
5.19.935	chart.series.i[].lineStyle.shadowOffsetY.....	2007
5.19.936	chart.series.i[].lineStyle.type.....	2008
	Erlaubte Werte .....	2008
5.19.937	chart.series.i[].lineStyle.width .....	2008
5.19.938	chart.series.i[].links.emphasis.label.align .....	2008
	Erlaubte Werte .....	2009
5.19.939	chart.series.i[].links.emphasis.label.backgroundColor .....	2009
5.19.940	chart.series.i[].links.emphasis.label.borderColor.....	2010



5.19.941	chart.series.i[].links.emphasis.label.borderRadius .....	2010
5.19.942	chart.series.i[].links.emphasis.label.borderWidth .....	2011
5.19.943	chart.series.i[].links.emphasis.label.color .....	2011
5.19.944	chart.series.i[].links.emphasis.label.fontFamily .....	2011
5.19.945	chart.series.i[].links.emphasis.label.fontSize .....	2012
5.19.946	chart.series.i[].links.emphasis.label.fontStyle .....	2012
	Erlaubte Werte .....	2012
5.19.947	chart.series.i[].links.emphasis.label.fontWeight .....	2012
	Erlaubte Werte .....	2013
5.19.948	chart.series.i[].links.emphasis.label.formatter .....	2013
5.19.949	chart.series.i[].links.emphasis.label.height .....	2014
5.19.950	chart.series.i[].links.emphasis.label.lineHeight .....	2015
5.19.951	chart.series.i[].links.emphasis.label.object .....	2015
5.19.952	chart.series.i[].links.emphasis.label.padding .....	2016
5.19.953	chart.series.i[].links.emphasis.label.position .....	2016
	Erlaubte Werte .....	2017
5.19.954	chart.series.i[].links.emphasis.label.rich.object .....	2017
5.19.955	chart.series.i[].links.emphasis.label.shadowBlur .....	2018
5.19.956	chart.series.i[].links.emphasis.label.shadowColor .....	2019
5.19.957	chart.series.i[].links.emphasis.label.shadowOffsetX .....	2019
5.19.958	chart.series.i[].links.emphasis.label.shadowOffsetY .....	2019
5.19.959	chart.series.i[].links.emphasis.label.show .....	2020
5.19.960	chart.series.i[].links.emphasis.label.textBorderColor .....	2020
5.19.961	chart.series.i[].links.emphasis.label.textBorderWidth .....	2020
5.19.962	chart.series.i[].links.emphasis.label.textShadowBlur .....	2021
5.19.963	chart.series.i[].links.emphasis.label.textShadowColor .....	2021
5.19.964	chart.series.i[].links.emphasis.label.textShadowOffsetX .....	2021
5.19.965	chart.series.i[].links.emphasis.label.textShadowOffsetY .....	2022
5.19.966	chart.series.i[].links.emphasis.label.verticalAlign .....	2022
	Erlaubte Werte .....	2023
5.19.967	chart.series.i[].links.emphasis.label.width .....	2023
5.19.968	chart.series.i[].links.emphasis.lineStyle.color .....	2023
5.19.969	chart.series.i[].links.emphasis.lineStyle.object .....	2025
5.19.970	chart.series.i[].links.emphasis.lineStyle.opacity .....	2025
5.19.971	chart.series.i[].links.emphasis.lineStyle.shadowBlur .....	2025
5.19.972	chart.series.i[].links.emphasis.lineStyle.shadowColor .....	2026

5.19.973	chart.series.i[].links.emphasis.lineStyle.shadowOffsetX	2026
5.19.974	chart.series.i[].links.emphasis.lineStyle.shadowOffsetY	2026
5.19.975	chart.series.i[].links.emphasis.lineStyle.type	2027
	Erlaubte Werte	2027
5.19.976	chart.series.i[].links.emphasis.lineStyle.width	2027
5.19.977	chart.series.i[].links.emphasis.object	2028
5.19.978	chart.series.i[].links.label.align	2028
	Erlaubte Werte	2028
5.19.979	chart.series.i[].links.label.backgroundColor	2029
5.19.980	chart.series.i[].links.label.borderColor	2029
5.19.981	chart.series.i[].links.label.borderRadius	2030
5.19.982	chart.series.i[].links.label.borderWidth	2030
5.19.983	chart.series.i[].links.label.color	2030
5.19.984	chart.series.i[].links.label.fontFamily	2031
5.19.985	chart.series.i[].links.label.fontSize	2031
5.19.986	chart.series.i[].links.label.fontStyle	2031
	Erlaubte Werte	2032
5.19.987	chart.series.i[].links.label.fontWeight	2032
	Erlaubte Werte	2032
5.19.988	chart.series.i[].links.label.formatter	2033
5.19.989	chart.series.i[].links.label.height	2034
5.19.990	chart.series.i[].links.label.lineHeight	2034
5.19.991	chart.series.i[].links.label.object	2035
5.19.992	chart.series.i[].links.label.padding	2035
5.19.993	chart.series.i[].links.label.position	2036
	Erlaubte Werte	2036
5.19.994	chart.series.i[].links.label.rich.object	2037
5.19.995	chart.series.i[].links.label.shadowBlur	2038
5.19.996	chart.series.i[].links.label.shadowColor	2039
5.19.997	chart.series.i[].links.label.shadowOffsetX	2039
5.19.998	chart.series.i[].links.label.shadowOffsetY	2039
5.19.999	chart.series.i[].links.label.show	2040
5.19.1000	chart.series.i[].links.label.textBorderColor	2040
5.19.1001	chart.series.i[].links.label.textBorderWidth	2040
5.19.1002	chart.series.i[].links.label.textShadowBlur	2041
5.19.1003	chart.series.i[].links.label.textShadowColor	2041

5.19.1004	chart.series.i[].links.label.textShadowOffsetX .....	2041
5.19.1005	chart.series.i[].links.label.textShadowOffsetY .....	2042
5.19.1006	chart.series.i[].links.label.verticalAlign .....	2042
	Erlaubte Werte .....	2043
5.19.1007	chart.series.i[].links.label.width .....	2043
5.19.1008	chart.series.i[].links.lineStyle.color .....	2043
5.19.1009	chart.series.i[].links.lineStyle.curveness .....	2045
5.19.1010	chart.series.i[].links.lineStyle.object .....	2045
5.19.1011	chart.series.i[].links.lineStyle.opacity .....	2045
5.19.1012	chart.series.i[].links.lineStyle.shadowBlur .....	2046
5.19.1013	chart.series.i[].links.lineStyle.shadowColor .....	2046
5.19.1014	chart.series.i[].links.lineStyle.shadowOffsetX .....	2046
5.19.1015	chart.series.i[].links.lineStyle.shadowOffsetY .....	2047
5.19.1016	chart.series.i[].links.lineStyle.type .....	2047
	Erlaubte Werte .....	2047
5.19.1017	chart.series.i[].links.lineStyle.width .....	2048
5.19.1018	chart.series.i[].links.object .....	2048
5.19.1019	chart.series.i[].links.source .....	2048
5.19.1020	chart.series.i[].links.symbol .....	2049
5.19.1021	chart.series.i[].links.symbolSize .....	2049
5.19.1022	chart.series.i[].links.target .....	2049
5.19.1023	chart.series.i[].links.value .....	2050
5.19.1024	chart.series.i[].markArea.animation .....	2050
5.19.1025	chart.series.i[].markArea.animationDelay .....	2051
5.19.1026	chart.series.i[].markArea.animationDelayUpdate .....	2051
5.19.1027	chart.series.i[].markArea.animationDuration .....	2052
5.19.1028	chart.series.i[].markArea.animationDurationUpdate .....	2052
5.19.1029	chart.series.i[].markArea.animationEasing .....	2053
	Erlaubte Werte .....	2053
5.19.1030	chart.series.i[].markArea.animationEasingUpdate .....	2054
	Erlaubte Werte .....	2054
5.19.1031	chart.series.i[].markArea.animationThreshold .....	2055
5.19.1032	chart.series.i[].markArea.data.object .....	2055
5.19.1033	chart.series.i[].markArea.itemStyle.borderColor .....	2057
5.19.1034	chart.series.i[].markArea.itemStyle.borderType .....	2057
	Erlaubte Werte .....	2057

5.19.1035	chart.series.i[].markArea.itemStyle.borderWidth	2057
5.19.1036	chart.series.i[].markArea.itemStyle.color	2058
5.19.1037	chart.series.i[].markArea.itemStyle.emphasis.borderColor	2059
5.19.1038	chart.series.i[].markArea.itemStyle.emphasis.borderType	2060
	Erlaubte Werte	2060
5.19.1039	chart.series.i[].markArea.itemStyle.emphasis.borderWidth	2060
5.19.1040	chart.series.i[].markArea.itemStyle.emphasis.color	2061
5.19.1041	chart.series.i[].markArea.itemStyle.emphasis.object	2062
5.19.1042	chart.series.i[].markArea.itemStyle.emphasis.opacity	2063
5.19.1043	chart.series.i[].markArea.itemStyle.emphasis.shadowBlur	2063
5.19.1044	chart.series.i[].markArea.itemStyle.emphasis.shadowColor	2064
5.19.1045	chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetX	2064
5.19.1046	chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetY	2064
5.19.1047	chart.series.i[].markArea.itemStyle.object	2065
5.19.1048	chart.series.i[].markArea.itemStyle.opacity	2065
5.19.1049	chart.series.i[].markArea.itemStyle.shadowBlur	2065
5.19.1050	chart.series.i[].markArea.itemStyle.shadowColor	2066
5.19.1051	chart.series.i[].markArea.itemStyle.shadowOffsetX	2066
5.19.1052	chart.series.i[].markArea.itemStyle.shadowOffsetY	2066
5.19.1053	chart.series.i[].markArea.label.align	2067
	Erlaubte Werte	2067
5.19.1054	chart.series.i[].markArea.label.backgroundColor	2068
5.19.1055	chart.series.i[].markArea.label.borderColor	2068
5.19.1056	chart.series.i[].markArea.label.borderRadius	2069
5.19.1057	chart.series.i[].markArea.label.borderWidth	2069
5.19.1058	chart.series.i[].markArea.label.color	2069
5.19.1059	chart.series.i[].markArea.label.distance	2070
5.19.1060	chart.series.i[].markArea.label.emphasis.align	2070
	Erlaubte Werte	2071
5.19.1061	chart.series.i[].markArea.label.emphasis.backgroundColor	2071
5.19.1062	chart.series.i[].markArea.label.emphasis.borderColor	2072
5.19.1063	chart.series.i[].markArea.label.emphasis.borderRadius	2072
5.19.1064	chart.series.i[].markArea.label.emphasis.borderWidth	2072
5.19.1065	chart.series.i[].markArea.label.emphasis.color	2073
5.19.1066	chart.series.i[].markArea.label.emphasis.distance	2073
5.19.1067	chart.series.i[].markArea.label.emphasis.fontFamily	2073

5.19.1068	chart.series.i[].markArea.label.emphasis.fontSize .....	2074
5.19.1069	chart.series.i[].markArea.label.emphasis.fontStyle.....	2074
	Erlaubte Werte .....	2075
5.19.1070	chart.series.i[].markArea.label.emphasis.fontWeight .....	2075
	Erlaubte Werte .....	2075
5.19.1071	chart.series.i[].markArea.label.emphasis.height .....	2075
5.19.1072	chart.series.i[].markArea.label.emphasis.lineHeight .....	2076
5.19.1073	chart.series.i[].markArea.label.emphasis.object .....	2076
5.19.1074	chart.series.i[].markArea.label.emphasis.offset .....	2077
5.19.1075	chart.series.i[].markArea.label.emphasis.padding.....	2077
5.19.1076	chart.series.i[].markArea.label.emphasis.position .....	2077
	Erlaubte Werte .....	2078
5.19.1077	chart.series.i[].markArea.label.emphasis.rich.object.....	2078
5.19.1078	chart.series.i[].markArea.label.emphasis.rotate.....	2079
5.19.1079	chart.series.i[].markArea.label.emphasis.shadowBlur.....	2080
5.19.1080	chart.series.i[].markArea.label.emphasis.shadowColor.....	2080
5.19.1081	chart.series.i[].markArea.label.emphasis.shadowOffsetX.....	2080
5.19.1082	chart.series.i[].markArea.label.emphasis.shadowOffsetY.....	2081
5.19.1083	chart.series.i[].markArea.label.emphasis.show .....	2081
5.19.1084	chart.series.i[].markArea.label.emphasis.textBorderColor.....	2081
5.19.1085	chart.series.i[].markArea.label.emphasis.textBorderWidth.....	2082
5.19.1086	chart.series.i[].markArea.label.emphasis.textShadowBlur.....	2082
5.19.1087	chart.series.i[].markArea.label.emphasis.textShadowColor.....	2083
5.19.1088	chart.series.i[].markArea.label.emphasis.textShadowOffsetX.....	2083
5.19.1089	chart.series.i[].markArea.label.emphasis.textShadowOffsetY.....	2083
5.19.1090	chart.series.i[].markArea.label.emphasis.verticalAlign .....	2084
	Erlaubte Werte .....	2084
5.19.1091	chart.series.i[].markArea.label.emphasis.width .....	2084
5.19.1092	chart.series.i[].markArea.label.fontFamily .....	2085
5.19.1093	chart.series.i[].markArea.label.fontSize .....	2085
5.19.1094	chart.series.i[].markArea.label.fontStyle.....	2086
	Erlaubte Werte .....	2086
5.19.1095	chart.series.i[].markArea.label.fontWeight .....	2086
	Erlaubte Werte .....	2086
5.19.1096	chart.series.i[].markArea.label.height .....	2087
5.19.1097	chart.series.i[].markArea.label.lineHeight.....	2087

5.19.1098	chart.series.i[].markArea.label.object .....	2088
5.19.1099	chart.series.i[].markArea.label.offset .....	2088
5.19.1100	chart.series.i[].markArea.label.padding .....	2088
5.19.1101	chart.series.i[].markArea.label.position .....	2089
	Erlaubte Werte .....	2089
5.19.1102	chart.series.i[].markArea.label.rich.object .....	2090
5.19.1103	chart.series.i[].markArea.label.rotate .....	2091
5.19.1104	chart.series.i[].markArea.label.shadowBlur .....	2092
5.19.1105	chart.series.i[].markArea.label.shadowColor .....	2092
5.19.1106	chart.series.i[].markArea.label.shadowOffsetX .....	2092
5.19.1107	chart.series.i[].markArea.label.shadowOffsetY .....	2093
5.19.1108	chart.series.i[].markArea.label.show .....	2093
5.19.1109	chart.series.i[].markArea.label.textBorderColor .....	2093
5.19.1110	chart.series.i[].markArea.label.textBorderWidth .....	2094
5.19.1111	chart.series.i[].markArea.label.textShadowBlur .....	2094
5.19.1112	chart.series.i[].markArea.label.textShadowColor .....	2094
5.19.1113	chart.series.i[].markArea.label.textShadowOffsetX .....	2095
5.19.1114	chart.series.i[].markArea.label.textShadowOffsetY .....	2095
5.19.1115	chart.series.i[].markArea.label.verticalAlign .....	2096
	Erlaubte Werte .....	2096
5.19.1116	chart.series.i[].markArea.label.width .....	2096
5.19.1117	chart.series.i[].markArea.object .....	2097
5.19.1118	chart.series.i[].markArea.silent .....	2097
5.19.1119	chart.series.i[].markLine.animation .....	2097
5.19.1120	chart.series.i[].markLine.animationDelay .....	2098
5.19.1121	chart.series.i[].markLine.animationDelayUpdate .....	2099
5.19.1122	chart.series.i[].markLine.animationDuration .....	2099
5.19.1123	chart.series.i[].markLine.animationDurationUpdate .....	2100
5.19.1124	chart.series.i[].markLine.animationEasing .....	2100
	Erlaubte Werte .....	2100
5.19.1125	chart.series.i[].markLine.animationEasingUpdate .....	2101
	Erlaubte Werte .....	2101
5.19.1126	chart.series.i[].markLine.animationThreshold .....	2102
5.19.1127	chart.series.i[].markLine.label.emphasis.formatter .....	2102
5.19.1128	chart.series.i[].markLine.label.emphasis.object .....	2104
5.19.1129	chart.series.i[].markLine.label.emphasis.position .....	2104

Erlaubte Werte .....	2104
5.19.1130 chart.series.i[].markLine.label.emphasis.show .....	2104
5.19.1131 chart.series.i[].markLine.label.formatter .....	2105
5.19.1132 chart.series.i[].markLine.label.object .....	2106
5.19.1133 chart.series.i[].markLine.label.position .....	2106
Erlaubte Werte .....	2107
5.19.1134 chart.series.i[].markLine.label.show .....	2107
5.19.1135 chart.series.i[].markLine.lineStyle.color .....	2107
5.19.1136 chart.series.i[].markLine.lineStyle.curveness .....	2108
5.19.1137 chart.series.i[].markLine.lineStyle.emphasis.color .....	2109
5.19.1138 chart.series.i[].markLine.lineStyle.emphasis.object .....	2110
5.19.1139 chart.series.i[].markLine.lineStyle.emphasis.opacity .....	2111
5.19.1140 chart.series.i[].markLine.lineStyle.emphasis.shadowBlur .....	2111
5.19.1141 chart.series.i[].markLine.lineStyle.emphasis.shadowColor .....	2112
5.19.1142 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetX .....	2112
5.19.1143 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetY .....	2112
5.19.1144 chart.series.i[].markLine.lineStyle.emphasis.type .....	2113
Erlaubte Werte .....	2113
5.19.1145 chart.series.i[].markLine.lineStyle.emphasis.width .....	2113
5.19.1146 chart.series.i[].markLine.lineStyle.object .....	2113
5.19.1147 chart.series.i[].markLine.lineStyle.opacity .....	2114
5.19.1148 chart.series.i[].markLine.lineStyle.shadowBlur .....	2114
5.19.1149 chart.series.i[].markLine.lineStyle.shadowColor .....	2115
5.19.1150 chart.series.i[].markLine.lineStyle.shadowOffsetX .....	2115
5.19.1151 chart.series.i[].markLine.lineStyle.shadowOffsetY .....	2115
5.19.1152 chart.series.i[].markLine.lineStyle.type .....	2116
Erlaubte Werte .....	2116
5.19.1153 chart.series.i[].markLine.lineStyle.width .....	2116
5.19.1154 chart.series.i[].markLine.object .....	2117
5.19.1155 chart.series.i[].markLine.precision .....	2117
5.19.1156 chart.series.i[].markLine.silent .....	2117
5.19.1157 chart.series.i[].markLine.symbol .....	2118
5.19.1158 chart.series.i[].markLine.symbolSize .....	2118
5.19.1159 chart.series.i[].markPoint.animation .....	2118
5.19.1160 chart.series.i[].markPoint.animationDelay .....	2119
5.19.1161 chart.series.i[].markPoint.animationDelayUpdate .....	2119

5.19.1162	chart.series.i[].markPoint.animationDuration	2120
5.19.1163	chart.series.i[].markPoint.animationDurationUpdate	2120
5.19.1164	chart.series.i[].markPoint.animationEasing	2121
	Erlaubte Werte	2121
5.19.1165	chart.series.i[].markPoint.animationEasingUpdate	2122
	Erlaubte Werte	2122
5.19.1166	chart.series.i[].markPoint.animationThreshold	2123
5.19.1167	chart.series.i[].markPoint.itemStyle.borderColor	2123
5.19.1168	chart.series.i[].markPoint.itemStyle.borderType	2123
	Erlaubte Werte	2124
5.19.1169	chart.series.i[].markPoint.itemStyle.borderWidth	2124
5.19.1170	chart.series.i[].markPoint.itemStyle.color	2124
5.19.1171	chart.series.i[].markPoint.itemStyle.emphasis.borderColor	2126
5.19.1172	chart.series.i[].markPoint.itemStyle.emphasis.borderType	2126
	Erlaubte Werte	2126
5.19.1173	chart.series.i[].markPoint.itemStyle.emphasis.borderWidth	2126
5.19.1174	chart.series.i[].markPoint.itemStyle.emphasis.color	2127
5.19.1175	chart.series.i[].markPoint.itemStyle.emphasis.object	2128
5.19.1176	chart.series.i[].markPoint.itemStyle.emphasis.opacity	2129
5.19.1177	chart.series.i[].markPoint.itemStyle.emphasis.shadowBlur	2129
5.19.1178	chart.series.i[].markPoint.itemStyle.emphasis.shadowColor	2130
5.19.1179	chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetX	2130
5.19.1180	chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetY	2130
5.19.1181	chart.series.i[].markPoint.itemStyle.object	2131
5.19.1182	chart.series.i[].markPoint.itemStyle.opacity	2131
5.19.1183	chart.series.i[].markPoint.itemStyle.shadowBlur	2131
5.19.1184	chart.series.i[].markPoint.itemStyle.shadowColor	2132
5.19.1185	chart.series.i[].markPoint.itemStyle.shadowOffsetX	2132
5.19.1186	chart.series.i[].markPoint.itemStyle.shadowOffsetY	2132
5.19.1187	chart.series.i[].markPoint.label.align	2133
	Erlaubte Werte	2133
5.19.1188	chart.series.i[].markPoint.label.backgroundColor	2134
5.19.1189	chart.series.i[].markPoint.label.borderColor	2134
5.19.1190	chart.series.i[].markPoint.label.borderRadius	2135
5.19.1191	chart.series.i[].markPoint.label.borderWidth	2135
5.19.1192	chart.series.i[].markPoint.label.color	2135



5.19.1193	chart.series.i[].markPoint.label.distance .....	2136
5.19.1194	chart.series.i[].markPoint.label.emphasis.align .....	2136
	Erlaubte Werte .....	2137
5.19.1195	chart.series.i[].markPoint.label.emphasis.backgroundColor .....	2137
5.19.1196	chart.series.i[].markPoint.label.emphasis.borderColor .....	2138
5.19.1197	chart.series.i[].markPoint.label.emphasis.borderRadius.....	2138
5.19.1198	chart.series.i[].markPoint.label.emphasis.borderWidth .....	2138
5.19.1199	chart.series.i[].markPoint.label.emphasis.color .....	2139
5.19.1200	chart.series.i[].markPoint.label.emphasis.distance .....	2139
5.19.1201	chart.series.i[].markPoint.label.emphasis.fontFamily .....	2139
5.19.1202	chart.series.i[].markPoint.label.emphasis.fontSize.....	2140
5.19.1203	chart.series.i[].markPoint.label.emphasis.fontStyle .....	2140
	Erlaubte Werte .....	2141
5.19.1204	chart.series.i[].markPoint.label.emphasis.fontWeight.....	2141
	Erlaubte Werte .....	2141
5.19.1205	chart.series.i[].markPoint.label.emphasis.formatter .....	2141
5.19.1206	chart.series.i[].markPoint.label.emphasis.height.....	2142
5.19.1207	chart.series.i[].markPoint.label.emphasis.lineHeight .....	2143
5.19.1208	chart.series.i[].markPoint.label.emphasis.object.....	2143
5.19.1209	chart.series.i[].markPoint.label.emphasis.padding.....	2144
5.19.1210	chart.series.i[].markPoint.label.emphasis.position.....	2144
	Erlaubte Werte .....	2145
5.19.1211	chart.series.i[].markPoint.label.emphasis.rich.object .....	2145
5.19.1212	chart.series.i[].markPoint.label.emphasis.rotate .....	2146
5.19.1213	chart.series.i[].markPoint.label.emphasis.shadowBlur .....	2147
5.19.1214	chart.series.i[].markPoint.label.emphasis.shadowColor .....	2147
5.19.1215	chart.series.i[].markPoint.label.emphasis.shadowOffsetX .....	2147
5.19.1216	chart.series.i[].markPoint.label.emphasis.shadowOffsetY .....	2148
5.19.1217	chart.series.i[].markPoint.label.emphasis.show.....	2148
5.19.1218	chart.series.i[].markPoint.label.emphasis.textBorderColor .....	2148
5.19.1219	chart.series.i[].markPoint.label.emphasis.textBorderWidth .....	2149
5.19.1220	chart.series.i[].markPoint.label.emphasis.textShadowBlur .....	2149
5.19.1221	chart.series.i[].markPoint.label.emphasis.textShadowColor .....	2149
5.19.1222	chart.series.i[].markPoint.label.emphasis.textShadowOffsetX .....	2150
5.19.1223	chart.series.i[].markPoint.label.emphasis.textShadowOffsetY .....	2150
5.19.1224	chart.series.i[].markPoint.label.emphasis.verticalAlign.....	2151

Erlaubte Werte .....	2151
5.19.1225 chart.series.i[].markPoint.label.emphasis.width .....	2151
5.19.1226 chart.series.i[].markPoint.label.fontFamily .....	2152
5.19.1227 chart.series.i[].markPoint.label.fontSize .....	2152
5.19.1228 chart.series.i[].markPoint.label.fontStyle .....	2153
Erlaubte Werte .....	2153
5.19.1229 chart.series.i[].markPoint.label.fontWeight .....	2153
Erlaubte Werte .....	2153
5.19.1230 chart.series.i[].markPoint.label.formatter .....	2154
5.19.1231 chart.series.i[].markPoint.label.height .....	2155
5.19.1232 chart.series.i[].markPoint.label.lineHeight .....	2155
5.19.1233 chart.series.i[].markPoint.label.object .....	2156
5.19.1234 chart.series.i[].markPoint.label.padding .....	2156
5.19.1235 chart.series.i[].markPoint.label.position .....	2157
Erlaubte Werte .....	2157
5.19.1236 chart.series.i[].markPoint.label.rich.object .....	2158
5.19.1237 chart.series.i[].markPoint.label.rotate .....	2159
5.19.1238 chart.series.i[].markPoint.label.shadowBlur .....	2160
5.19.1239 chart.series.i[].markPoint.label.shadowColor .....	2160
5.19.1240 chart.series.i[].markPoint.label.shadowOffsetX .....	2160
5.19.1241 chart.series.i[].markPoint.label.shadowOffsetY .....	2161
5.19.1242 chart.series.i[].markPoint.label.show .....	2161
5.19.1243 chart.series.i[].markPoint.label.textBorderColor .....	2161
5.19.1244 chart.series.i[].markPoint.label.textBorderWidth .....	2162
5.19.1245 chart.series.i[].markPoint.label.textShadowBlur .....	2162
5.19.1246 chart.series.i[].markPoint.label.textShadowColor .....	2162
5.19.1247 chart.series.i[].markPoint.label.textShadowOffsetX .....	2163
5.19.1248 chart.series.i[].markPoint.label.textShadowOffsetY .....	2163
5.19.1249 chart.series.i[].markPoint.label.verticalAlign .....	2164
Erlaubte Werte .....	2164
5.19.1250 chart.series.i[].markPoint.label.width .....	2164
5.19.1251 chart.series.i[].markPoint.object .....	2165
5.19.1252 chart.series.i[].markPoint.silent .....	2165
5.19.1253 chart.series.i[].markPoint.symbol .....	2165
Erlaubte Werte .....	2166
5.19.1254 chart.series.i[].markPoint.symbolKeepAspect .....	2166

5.19.1255	chart.series.i[].markPoint.symbolOffset .....	2167
	Erlaubte Werte .....	2167
5.19.1256	chart.series.i[].markPoint.symbolRotate .....	2167
5.19.1257	chart.series.i[].markPoint.symbolSize .....	2167
5.19.1258	chart.series.i[].max .....	2168
5.19.1259	chart.series.i[].maxOpacity .....	2168
5.19.1260	chart.series.i[].maxSize .....	2169
5.19.1261	chart.series.i[].min .....	2169
5.19.1262	chart.series.i[].minAngle .....	2169
5.19.1263	chart.series.i[].minOpacity .....	2170
5.19.1264	chart.series.i[].minSize .....	2170
5.19.1265	chart.series.i[].nameMap .....	2171
5.19.1266	chart.series.i[].nodeClick .....	2171
5.19.1267	chart.series.i[].nodeGap .....	2172
5.19.1268	chart.series.i[].nodes .....	2172
5.19.1269	chart.series.i[].nodeScaleRatio .....	2172
5.19.1270	chart.series.i[].nodeWidth .....	2173
5.19.1271	chart.series.i[].orient .....	2173
	Erlaubte Werte .....	2174
5.19.1272	chart.series.i[].parallellIndex .....	2174
5.19.1273	chart.series.i[].pointer.length .....	2174
	Erlaubte Werte .....	2174
5.19.1274	chart.series.i[].pointer.object .....	2175
5.19.1275	chart.series.i[].pointer.show .....	2175
5.19.1276	chart.series.i[].pointer.width .....	2175
5.19.1277	chart.series.i[].polarIndex .....	2176
5.19.1278	chart.series.i[].polyline .....	2176
5.19.1279	chart.series.i[].progressive .....	2176
5.19.1280	chart.series.i[].progressiveChunkMode .....	2177
	Erlaubte Werte .....	2177
5.19.1281	chart.series.i[].progressiveThreshold .....	2177
5.19.1282	chart.series.i[].radarIndex .....	2178
5.19.1283	chart.series.i[].radius .....	2178
	Erlaubte Werte .....	2179
5.19.1284	chart.series.i[].realtime .....	2179
5.19.1285	chart.series.i[].renderItem.object .....	2179

5.19.1286	chart.series.i[].renderLabelForZeroData .....	2181
5.19.1287	chart.series.i[].right.....	2181
	Erlaubte Werte .....	2181
5.19.1288	chart.series.i[].roam.....	2181
	Erlaubte Werte .....	2182
5.19.1289	chart.series.i[].roseType .....	2182
	Erlaubte Werte .....	2182
5.19.1290	chart.series.i[].sampling .....	2182
	Erlaubte Werte .....	2183
5.19.1291	chart.series.i[].selectedMode .....	2183
	Erlaubte Werte .....	2183
5.19.1292	chart.series.i[].selectedOffset .....	2184
5.19.1293	chart.series.i[].seriesLayoutBy .....	2184
	Erlaubte Werte .....	2184
5.19.1294	chart.series.i[].showAllSymbol.....	2185
	Erlaubte Werte .....	2185
5.19.1295	chart.series.i[].showEffectOn .....	2185
	Erlaubte Werte .....	2186
5.19.1296	chart.series.i[].showSymbol.....	2186
5.19.1297	chart.series.i[].silent .....	2186
5.19.1298	chart.series.i[].silent.link .....	2186
5.19.1299	chart.series.i[].silent.object.....	2187
5.19.1300	chart.series.i[].silent.target .....	2187
	Erlaubte Werte .....	2188
5.19.1301	chart.series.i[].silent.tooltip.backgroundColor.....	2188
5.19.1302	chart.series.i[].silent.tooltip.borderColor.....	2188
5.19.1303	chart.series.i[].silent.tooltip.borderWidth.....	2188
5.19.1304	chart.series.i[].silent.tooltip.extraCSSText.....	2189
5.19.1305	chart.series.i[].silent.tooltip.formatter .....	2189
5.19.1306	chart.series.i[].silent.tooltip.object .....	2192
5.19.1307	chart.series.i[].silent.tooltip.padding .....	2192
5.19.1308	chart.series.i[].silent.tooltip.position .....	2193
5.19.1309	chart.series.i[].silent.tooltip.textStyle.color.....	2194
5.19.1310	chart.series.i[].silent.tooltip.textStyle.fontFamily .....	2195
5.19.1311	chart.series.i[].silent.tooltip.textStyle.fontSize .....	2195
5.19.1312	chart.series.i[].silent.tooltip.textStyle.fontStyle.....	2196

Erlaubte Werte .....	2196
5.19.1313 chart.series.i[].silent.tooltip.textStyle.fontWeight.....	2196
Erlaubte Werte .....	2196
5.19.1314 chart.series.i[].silent.tooltip.textStyle.height .....	2197
5.19.1315 chart.series.i[].silent.tooltip.textStyle.lineHeight.....	2197
5.19.1316 chart.series.i[].silent.tooltip.textStyle.object.....	2198
5.19.1317 chart.series.i[].silent.tooltip.textStyle.textBorderColor .....	2198
5.19.1318 chart.series.i[].silent.tooltip.textStyle.textBorderWidth .....	2198
5.19.1319 chart.series.i[].silent.tooltip.textStyle.textShadowBlur .....	2199
5.19.1320 chart.series.i[].silent.tooltip.textStyle.textShadowColor .....	2199
5.19.1321 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetX.....	2200
5.19.1322 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetY .....	2200
5.19.1323 chart.series.i[].silent.tooltip.textStyle.width .....	2200
5.19.1324 chart.series.i[].singleAxisIndex.....	2201
5.19.1325 chart.series.i[].smooth.....	2201
5.19.1326 chart.series.i[].smoothMonotone .....	2202
Erlaubte Werte .....	2202
5.19.1327 chart.series.i[].sort.....	2202
5.19.1328 chart.series.i[].splitLine.length .....	2203
Erlaubte Werte .....	2203
5.19.1329 chart.series.i[].splitLine.lineStyle.color .....	2203
5.19.1330 chart.series.i[].splitLine.lineStyle.object .....	2205
5.19.1331 chart.series.i[].splitLine.lineStyle.opacity .....	2205
5.19.1332 chart.series.i[].splitLine.lineStyle.shadowBlur .....	2205
5.19.1333 chart.series.i[].splitLine.lineStyle.shadowColor .....	2206
5.19.1334 chart.series.i[].splitLine.lineStyle.shadowOffsetX .....	2206
5.19.1335 chart.series.i[].splitLine.lineStyle.shadowOffsetY .....	2206
5.19.1336 chart.series.i[].splitLine.lineStyle.type .....	2207
Erlaubte Werte .....	2207
5.19.1337 chart.series.i[].splitLine.lineStyle.width .....	2207
5.19.1338 chart.series.i[].splitLine.object.....	2208
5.19.1339 chart.series.i[].splitLine.show .....	2208
5.19.1340 chart.series.i[].splitNumber.....	2208
5.19.1341 chart.series.i[].squareRatio .....	2209
5.19.1342 chart.series.i[].stack.....	2209
5.19.1343 chart.series.i[].startAngle .....	2209

5.19.1344	chart.series.i[].step .....	2210
	Erlaubte Werte .....	2210
5.19.1345	chart.series.i[].stillShowZeroSum .....	2210
5.19.1346	chart.series.i[].symbol .....	2211
	Erlaubte Werte .....	2211
5.19.1347	chart.series.i[].symbolBoundingData .....	2211
5.19.1348	chart.series.i[].symbolClip .....	2213
5.19.1349	chart.series.i[].symbolKeepAspect .....	2213
5.19.1350	chart.series.i[].symbolMargin .....	2214
5.19.1351	chart.series.i[].symbolOffset .....	2215
	Erlaubte Werte .....	2215
5.19.1352	chart.series.i[].symbolPatternSize .....	2215
5.19.1353	chart.series.i[].symbolRepeat .....	2216
	Erlaubte Werte .....	2217
5.19.1354	chart.series.i[].symbolRepeatDirection .....	2218
	Erlaubte Werte .....	2218
5.19.1355	chart.series.i[].symbolRotate .....	2219
5.19.1356	chart.series.i[].symbolSize .....	2219
5.19.1357	chart.series.i[].title.backgroundColor .....	2220
5.19.1358	chart.series.i[].title.borderColor .....	2221
5.19.1359	chart.series.i[].title.borderRadius .....	2221
5.19.1360	chart.series.i[].title.borderWidth .....	2222
5.19.1361	chart.series.i[].title.color .....	2222
5.19.1362	chart.series.i[].title.fontFamily .....	2222
5.19.1363	chart.series.i[].title.fontSize .....	2223
5.19.1364	chart.series.i[].title.fontStyle .....	2223
	Erlaubte Werte .....	2224
5.19.1365	chart.series.i[].title.fontWeight .....	2224
	Erlaubte Werte .....	2224
5.19.1366	chart.series.i[].title.height .....	2224
5.19.1367	chart.series.i[].title.lineHeight .....	2225
5.19.1368	chart.series.i[].title.object .....	2225
5.19.1369	chart.series.i[].title.offsetCenter .....	2226
	Erlaubte Werte .....	2226
5.19.1370	chart.series.i[].title.padding .....	2226
5.19.1371	chart.series.i[].title.rich.object .....	2227

5.19.1372	chart.series.i[].title.shadowBlur	2228
5.19.1373	chart.series.i[].title.shadowColor	2228
5.19.1374	chart.series.i[].title.shadowOffsetX	2228
5.19.1375	chart.series.i[].title.shadowOffsetY	2229
5.19.1376	chart.series.i[].title.show	2229
5.19.1377	chart.series.i[].title.textBorderColor	2229
5.19.1378	chart.series.i[].title.textBorderWidth	2230
5.19.1379	chart.series.i[].title.textShadowBlur	2230
5.19.1380	chart.series.i[].title.textShadowColor	2230
5.19.1381	chart.series.i[].title.textShadowOffsetX	2231
5.19.1382	chart.series.i[].title.textShadowOffsetY	2231
5.19.1383	chart.series.i[].title.width	2232
5.19.1384	chart.series.i[].tooltip.backgroundColor	2232
5.19.1385	chart.series.i[].tooltip.borderColor	2232
5.19.1386	chart.series.i[].tooltip.borderWidth	2233
5.19.1387	chart.series.i[].tooltip.extraCssText	2233
5.19.1388	chart.series.i[].tooltip.formatter	2234
5.19.1389	chart.series.i[].tooltip.object	2236
5.19.1390	chart.series.i[].tooltip.padding	2236
5.19.1391	chart.series.i[].tooltip.position	2237
5.19.1392	chart.series.i[].tooltip.textStyle.color	2238
5.19.1393	chart.series.i[].tooltip.textStyle.fontFamily	2239
5.19.1394	chart.series.i[].tooltip.textStyle.fontSize	2239
5.19.1395	chart.series.i[].tooltip.textStyle.fontStyle	2240
	Erlaubte Werte	2240
5.19.1396	chart.series.i[].tooltip.textStyle.fontWeight	2240
	Erlaubte Werte	2240
5.19.1397	chart.series.i[].tooltip.textStyle.height	2241
5.19.1398	chart.series.i[].tooltip.textStyle.lineHeight	2241
5.19.1399	chart.series.i[].tooltip.textStyle.object	2242
5.19.1400	chart.series.i[].tooltip.textStyle.textBorderColor	2242
5.19.1401	chart.series.i[].tooltip.textStyle.textBorderWidth	2242
5.19.1402	chart.series.i[].tooltip.textStyle.textShadowBlur	2243
5.19.1403	chart.series.i[].tooltip.textStyle.textShadowColor	2243
5.19.1404	chart.series.i[].tooltip.textStyle.textShadowOffsetX	2244
5.19.1405	chart.series.i[].tooltip.textStyle.textShadowOffsetY	2244

5.19.1406	chart.series.i[].tooltip.textStyle.width	2244
5.19.1407	chart.series.i[].top	2245
	Erlaubte Werte	2245
5.19.1408	chart.series.i[].upperLabel.align	2245
	Erlaubte Werte	2246
5.19.1409	chart.series.i[].upperLabel.backgroundColor	2246
5.19.1410	chart.series.i[].upperLabel.borderColor	2247
5.19.1411	chart.series.i[].upperLabel.borderRadius	2247
5.19.1412	chart.series.i[].upperLabel.borderWidth	2248
5.19.1413	chart.series.i[].upperLabel.color	2248
5.19.1414	chart.series.i[].upperLabel.distance	2248
5.19.1415	chart.series.i[].upperLabel.ellipsis	2249
5.19.1416	chart.series.i[].upperLabel.fontFamily	2249
5.19.1417	chart.series.i[].upperLabel.fontSize	2249
5.19.1418	chart.series.i[].upperLabel.fontStyle	2250
	Erlaubte Werte	2250
5.19.1419	chart.series.i[].upperLabel.fontWeight	2250
	Erlaubte Werte	2251
5.19.1420	chart.series.i[].upperLabel.formatter	2251
5.19.1421	chart.series.i[].upperLabel.height	2252
5.19.1422	chart.series.i[].upperLabel.lineHeight	2252
5.19.1423	chart.series.i[].upperLabel.object	2253
5.19.1424	chart.series.i[].upperLabel.offset	2254
5.19.1425	chart.series.i[].upperLabel.padding	2254
5.19.1426	chart.series.i[].upperLabel.position	2254
	Erlaubte Werte	2255
5.19.1427	chart.series.i[].upperLabel.rich.object	2255
5.19.1428	chart.series.i[].upperLabel.rotate	2256
5.19.1429	chart.series.i[].upperLabel.shadowBlur	2257
5.19.1430	chart.series.i[].upperLabel.shadowColor	2257
5.19.1431	chart.series.i[].upperLabel.shadowOffsetX	2257
5.19.1432	chart.series.i[].upperLabel.shadowOffsetY	2258
5.19.1433	chart.series.i[].upperLabel.show	2258
5.19.1434	chart.series.i[].upperLabel.textBorderColor	2258
5.19.1435	chart.series.i[].upperLabel.textBorderWidth	2259
5.19.1436	chart.series.i[].upperLabel.textShadowBlur	2259



5.19.1437	chart.series.i[].upperLabel.textShadowColor .....	2259
5.19.1438	chart.series.i[].upperLabel.textShadowOffsetX .....	2260
5.19.1439	chart.series.i[].upperLabel.textShadowOffsetY .....	2260
5.19.1440	chart.series.i[].upperLabel.verticalAlign .....	2261
	Erlaubte Werte .....	2261
5.19.1441	chart.series.i[].upperLabel.width.....	2261
5.19.1442	chart.series.i[].visibleMin.....	2262
5.19.1443	chart.series.i[].visualDimension.....	2262
5.19.1444	chart.series.i[].visualMax .....	2263
5.19.1445	chart.series.i[].visualMin.....	2263
5.19.1446	chart.series.i[].width .....	2264
5.19.1447	chart.series.i[].xAxisIndex.....	2264
5.19.1448	chart.series.i[].yAxisIndex.....	2264
5.19.1449	chart.series.i[].z.....	2265
5.19.1450	chart.series.i[].zlevel .....	2265
5.19.1451	chart.series.i[].zoom .....	2266
5.19.1452	chart.series.i[].zoomToNodeRatio .....	2266
5.19.1453	chart.series.i[].silent.children .....	2266
5.19.1454	chart.series.fixed.name .....	2267
5.19.1455	chart.series.i[].name .....	2267
5.19.1456	chart.series.i[].label.alignTo.....	2268
	Erlaubte Werte .....	2268
5.19.1457	chart.series.i[].label.edgeDistance .....	2268
5.19.1458	chart.series.i[].label.bleedMargin .....	2269
5.19.1459	chart.series.i[].label.distanceToLabelLine .....	2269
5.19.1460	chart.series.i[].minShowLabelAngle .....	2269
5.20	ECharts (4) Single Axis.....	2270
5.20.1	chart.singleAxis.object.....	2270
5.20.2	chart.singleAxis.id .....	2270
5.20.3	chart.singleAxis.zlevel.....	2270
5.20.4	chart.singleAxis.z .....	2271
5.20.5	chart.singleAxis.left.....	2271
	Erlaubte Werte .....	2272
5.20.6	chart.singleAxis.top .....	2272
	Erlaubte Werte .....	2272
5.20.7	chart.singleAxis.right .....	2273

	Erlaubte Werte .....	2273
5.20.8	chart.singleAxis.bottom.....	2273
	Erlaubte Werte .....	2273
5.20.9	chart.singleAxis.width.....	2274
5.20.10	chart.singleAxis.height .....	2274
5.20.11	chart.singleAxis.orient .....	2274
	Erlaubte Werte .....	2275
5.20.12	chart.singleAxis.type.....	2275
	Erlaubte Werte .....	2275
5.20.13	chart.singleAxis.name.....	2275
5.20.14	chart.singleAxis.nameLocation .....	2276
	Erlaubte Werte .....	2276
5.20.15	chart.singleAxis.nameTextStyle.object .....	2276
5.20.16	chart.singleAxis.nameTextStyle.color .....	2277
5.20.17	chart.singleAxis.nameTextStyle.fontStyle.....	2277
	Erlaubte Werte .....	2277
5.20.18	chart.singleAxis.nameTextStyle.fontWeight .....	2277
	Erlaubte Werte .....	2278
5.20.19	chart.singleAxis.nameTextStyle.fontFamily.....	2278
5.20.20	chart.singleAxis.nameTextStyle.fontSize .....	2278
5.20.21	chart.singleAxis.nameTextStyle.align.....	2279
	Erlaubte Werte .....	2279
5.20.22	chart.singleAxis.nameTextStyle.verticalAlign .....	2279
	Erlaubte Werte .....	2280
5.20.23	chart.singleAxis.nameTextStyle.lineHeight.....	2280
5.20.24	chart.singleAxis.nameTextStyle.backgroundColor.....	2281
5.20.25	chart.singleAxis.nameTextStyle.borderColor .....	2281
5.20.26	chart.singleAxis.nameTextStyle.borderWidth.....	2282
5.20.27	chart.singleAxis.nameTextStyle.borderRadius .....	2282
5.20.28	chart.singleAxis.nameTextStyle.padding .....	2282
5.20.29	chart.singleAxis.nameTextStyle.shadowColor.....	2283
5.20.30	chart.singleAxis.nameTextStyle.shadowBlur.....	2283
5.20.31	chart.singleAxis.nameTextStyle.shadowOffsetX.....	2284
5.20.32	chart.singleAxis.nameTextStyle.shadowOffsetY.....	2284
5.20.33	chart.singleAxis.nameTextStyle.width .....	2284
5.20.34	chart.singleAxis.nameTextStyle.height .....	2285

5.20.35	chart.singleAxis.nameTextStyle.textBorderColor .....	2285
5.20.36	chart.singleAxis.nameTextStyle.textBorderWidth .....	2286
5.20.37	chart.singleAxis.nameTextStyle.textShadowColor .....	2286
5.20.38	chart.singleAxis.nameTextStyle.textShadowBlur .....	2286
5.20.39	chart.singleAxis.nameTextStyle.textShadowOffsetX .....	2287
5.20.40	chart.singleAxis.nameTextStyle.textShadowOffsetY .....	2287
5.20.41	chart.singleAxis.nameTextStyle.rich.object .....	2287
5.20.42	chart.singleAxis.nameGap .....	2288
5.20.43	chart.singleAxis.nameRotate .....	2289
5.20.44	chart.singleAxis.inverse .....	2289
5.20.45	chart.singleAxis.boundaryGap .....	2289
5.20.46	chart.singleAxis.min .....	2290
5.20.47	chart.singleAxis.max .....	2291
5.20.48	chart.singleAxis.scale .....	2291
5.20.49	chart.singleAxis.splitNumber .....	2291
5.20.50	chart.singleAxis.minInterval .....	2292
5.20.51	chart.singleAxis.interval .....	2292
5.20.52	chart.singleAxis.logBase .....	2293
5.20.53	chart.singleAxis.silent .....	2293
5.20.54	chart.singleAxis.triggerEvent .....	2294
5.20.55	chart.singleAxis.axisLine.object .....	2294
5.20.56	chart.singleAxis.axisLine.show .....	2295
5.20.57	chart.singleAxis.axisLine.symbol .....	2295
	Erlaubte Werte .....	2295
5.20.58	chart.singleAxis.axisLine.symbolSize .....	2295
5.20.59	chart.singleAxis.axisLine.symbolOffset .....	2296
5.20.60	chart.singleAxis.axisLine.lineStyle.object .....	2296
5.20.61	chart.singleAxis.axisLine.lineStyle.color .....	2297
5.20.62	chart.singleAxis.axisLine.lineStyle.width .....	2298
5.20.63	chart.singleAxis.axisLine.lineStyle.type .....	2299
	Erlaubte Werte .....	2299
5.20.64	chart.singleAxis.axisLine.lineStyle.shadowBlur .....	2299
5.20.65	chart.singleAxis.axisLine.lineStyle.shadowColor .....	2300
5.20.66	chart.singleAxis.axisLine.lineStyle.shadowOffsetX .....	2300
5.20.67	chart.singleAxis.axisLine.lineStyle.shadowOffsetY .....	2300
5.20.68	chart.singleAxis.axisLine.lineStyle.opacity .....	2301

5.20.69	chart.singleAxis.axisTick.object .....	2301
5.20.70	chart.singleAxis.axisTick.show .....	2301
5.20.71	chart.singleAxis.axisTick.alignWithLabel.....	2302
5.20.72	chart.singleAxis.axisTick.interval.....	2302
5.20.73	chart.singleAxis.axisTick.inside.....	2303
5.20.74	chart.singleAxis.axisTick.length .....	2303
5.20.75	chart.singleAxis.axisTick.lineStyle.object.....	2303
5.20.76	chart.singleAxis.axisTick.lineStyle.color.....	2304
5.20.77	chart.singleAxis.axisTick.lineStyle.width .....	2304
5.20.78	chart.singleAxis.axisTick.lineStyle.type.....	2304
	Erlaubte Werte .....	2305
5.20.79	chart.singleAxis.axisTick.lineStyle.shadowBlur .....	2305
5.20.80	chart.singleAxis.axisTick.lineStyle.shadowColor .....	2305
5.20.81	chart.singleAxis.axisTick.lineStyle.shadowOffsetX.....	2306
5.20.82	chart.singleAxis.axisTick.lineStyle.shadowOffsetY .....	2306
5.20.83	chart.singleAxis.axisTick.lineStyle.opacity.....	2306
5.20.84	chart.singleAxis.axisLabel.object.....	2307
5.20.85	chart.singleAxis.axisLabel.show.....	2307
5.20.86	chart.singleAxis.axisLabel.interval.....	2307
5.20.87	chart.singleAxis.axisLabel.inside .....	2308
5.20.88	chart.singleAxis.axisLabel.rotate .....	2308
5.20.89	chart.singleAxis.axisLabel.margin.....	2308
5.20.90	chart.singleAxis.axisLabel.formatter .....	2309
5.20.91	chart.singleAxis.axisLabel.showMinLabel .....	2309
5.20.92	chart.singleAxis.axisLabel.showMaxLabel.....	2310
5.20.93	chart.singleAxis.axisLabel.color .....	2310
5.20.94	chart.singleAxis.axisLabel.fontStyle .....	2311
	Erlaubte Werte .....	2311
5.20.95	chart.singleAxis.axisLabel.fontWeight.....	2311
	Erlaubte Werte .....	2312
5.20.96	chart.singleAxis.axisLabel.fontFamily .....	2312
5.20.97	chart.singleAxis.axisLabel.fontSize.....	2312
5.20.98	chart.singleAxis.axisLabel.align .....	2313
	Erlaubte Werte .....	2313
5.20.99	chart.singleAxis.axisLabel.verticalAlign.....	2313
	Erlaubte Werte .....	2314

5.20.100	chart.singleAxis.axisLabel.lineHeight .....	2314
5.20.101	chart.singleAxis.axisLabel.backgroundColor .....	2315
5.20.102	chart.singleAxis.axisLabel.borderColor .....	2315
5.20.103	chart.singleAxis.axisLabel.borderWidth .....	2316
5.20.104	chart.singleAxis.axisLabel.borderRadius.....	2316
5.20.105	chart.singleAxis.axisLabel.padding.....	2316
5.20.106	chart.singleAxis.axisLabel.shadowColor .....	2317
5.20.107	chart.singleAxis.axisLabel.shadowBlur .....	2317
5.20.108	chart.singleAxis.axisLabel.shadowOffsetX .....	2318
5.20.109	chart.singleAxis.axisLabel.shadowOffsetY .....	2318
5.20.110	chart.singleAxis.axisLabel.width .....	2318
5.20.111	chart.singleAxis.axisLabel.height.....	2319
5.20.112	chart.singleAxis.axisLabel.textBorderColor .....	2319
5.20.113	chart.singleAxis.axisLabel.textBorderWidth .....	2320
5.20.114	chart.singleAxis.axisLabel.textShadowColor .....	2320
5.20.115	chart.singleAxis.axisLabel.textShadowBlur .....	2320
5.20.116	chart.singleAxis.axisLabel.textShadowOffsetX .....	2321
5.20.117	chart.singleAxis.axisLabel.textShadowOffsetY .....	2321
5.20.118	chart.singleAxis.axisLabel.rich.object .....	2321
5.20.119	chart.singleAxis.splitLine.object .....	2322
5.20.120	chart.singleAxis.splitLine.show .....	2323
5.20.121	chart.singleAxis.splitLine.interval.....	2323
5.20.122	chart.singleAxis.splitLine.lineStyle.object.....	2323
5.20.123	chart.singleAxis.splitLine.lineStyle.color.....	2324
5.20.124	chart.singleAxis.splitLine.lineStyle.width.....	2324
5.20.125	chart.singleAxis.splitLine.lineStyle.type.....	2325
	Erlaubte Werte .....	2325
5.20.126	chart.singleAxis.splitLine.lineStyle.shadowBlur .....	2325
5.20.127	chart.singleAxis.splitLine.lineStyle.shadowColor .....	2326
5.20.128	chart.singleAxis.splitLine.lineStyle.shadowOffsetX.....	2326
5.20.129	chart.singleAxis.splitLine.lineStyle.shadowOffsetY .....	2326
5.20.130	chart.singleAxis.splitLine.lineStyle.opacity.....	2327
5.20.131	chart.singleAxis.splitArea.object.....	2327
5.20.132	chart.singleAxis.splitArea.interval .....	2327
5.20.133	chart.singleAxis.splitArea.show .....	2328
5.20.134	chart.singleAxis.splitArea.areaStyle.object .....	2328

5.20.135	chart.singleAxis.splitArea.areaStyle.color .....	2328
5.20.136	chart.singleAxis.splitArea.areaStyle.shadowBlur .....	2329
5.20.137	chart.singleAxis.splitArea.areaStyle.shadowColor .....	2329
5.20.138	chart.singleAxis.splitArea.areaStyle.shadowOffsetX .....	2330
5.20.139	chart.singleAxis.splitArea.areaStyle.shadowOffsetY .....	2330
5.20.140	chart.singleAxis.splitArea.areaStyle.opacity .....	2330
5.20.141	chart.singleAxis.axisPointer.object.....	2331
5.20.142	chart.singleAxis.axisPointer.show .....	2331
5.20.143	chart.singleAxis.axisPointer.type.....	2332
	Erlaubte Werte .....	2332
5.20.144	chart.singleAxis.axisPointer.snap .....	2332
5.20.145	chart.singleAxis.axisPointer.z.....	2332
5.20.146	chart.singleAxis.axisPointer.label.object .....	2333
5.20.147	chart.singleAxis.axisPointer.label.show .....	2333
5.20.148	chart.singleAxis.axisPointer.label.precision.....	2334
5.20.149	chart.singleAxis.axisPointer.label.formatter.....	2334
5.20.150	chart.singleAxis.axisPointer.label.margin .....	2335
5.20.151	chart.singleAxis.axisPointer.label.color .....	2336
5.20.152	chart.singleAxis.axisPointer.label.fontStyle.....	2336
	Erlaubte Werte .....	2336
5.20.153	chart.singleAxis.axisPointer.label.fontWeight .....	2337
	Erlaubte Werte .....	2337
5.20.154	chart.singleAxis.axisPointer.label.fontFamily.....	2337
5.20.155	chart.singleAxis.axisPointer.label.fontSize .....	2338
5.20.156	chart.singleAxis.axisPointer.label.lineHeight.....	2338
5.20.157	chart.singleAxis.axisPointer.label.width .....	2339
5.20.158	chart.singleAxis.axisPointer.label.height .....	2339
5.20.159	chart.singleAxis.axisPointer.label.textBorderColor.....	2339
5.20.160	chart.singleAxis.axisPointer.label.textBorderWidth.....	2340
5.20.161	chart.singleAxis.axisPointer.label.textShadowColor.....	2340
5.20.162	chart.singleAxis.axisPointer.label.textShadowBlur.....	2341
5.20.163	chart.singleAxis.axisPointer.label.textShadowOffsetX.....	2341
5.20.164	chart.singleAxis.axisPointer.label.textShadowOffsetY.....	2341
5.20.165	chart.singleAxis.axisPointer.label.padding .....	2342
5.20.166	chart.singleAxis.axisPointer.label.backgroundColor.....	2342
5.20.167	chart.singleAxis.axisPointer.label.borderColor.....	2343

5.20.168	chart.singleAxis.axisPointer.label.borderWidth.....	2343
5.20.169	chart.singleAxis.axisPointer.label.shadowBlur.....	2343
5.20.170	chart.singleAxis.axisPointer.label.shadowColor.....	2344
5.20.171	chart.singleAxis.axisPointer.label.shadowOffsetX.....	2344
5.20.172	chart.singleAxis.axisPointer.label.shadowOffsetY.....	2345
5.20.173	chart.singleAxis.axisPointer.lineStyle.object.....	2345
5.20.174	chart.singleAxis.axisPointer.lineStyle.color.....	2345
5.20.175	chart.singleAxis.axisPointer.lineStyle.width.....	2347
5.20.176	chart.singleAxis.axisPointer.lineStyle.type.....	2347
	Erlaubte Werte.....	2347
5.20.177	chart.singleAxis.axisPointer.lineStyle.shadowBlur.....	2347
5.20.178	chart.singleAxis.axisPointer.lineStyle.shadowColor.....	2348
5.20.179	chart.singleAxis.axisPointer.lineStyle.shadowOffsetX.....	2348
5.20.180	chart.singleAxis.axisPointer.lineStyle.shadowOffsetY.....	2349
5.20.181	chart.singleAxis.axisPointer.lineStyle.opacity.....	2349
5.20.182	chart.singleAxis.axisPointer.shadowStyle.object.....	2349
5.20.183	chart.singleAxis.axisPointer.shadowStyle.color.....	2350
5.20.184	chart.singleAxis.axisPointer.shadowStyle.shadowBlur.....	2351
5.20.185	chart.singleAxis.axisPointer.shadowStyle.shadowColor.....	2352
5.20.186	chart.singleAxis.axisPointer.shadowStyle.shadowOffsetX.....	2352
5.20.187	chart.singleAxis.axisPointer.shadowStyle.shadowOffsetY.....	2353
5.20.188	chart.singleAxis.axisPointer.shadowStyle.opacity.....	2353
5.20.189	chart.singleAxis.axisPointer.triggerTooltip.....	2353
5.20.190	chart.singleAxis.axisPointer.value.....	2354
5.20.191	chart.singleAxis.axisPointer.status.....	2354
	Erlaubte Werte.....	2354
5.20.192	chart.singleAxis.axisPointer.handle.object.....	2354
5.20.193	chart.singleAxis.axisPointer.handle.show.....	2355
5.20.194	chart.singleAxis.axisPointer.handle.icon.....	2355
5.20.195	chart.singleAxis.axisPointer.handle.size.....	2356
5.20.196	chart.singleAxis.axisPointer.handle.margin.....	2356
5.20.197	chart.singleAxis.axisPointer.handle.color.....	2356
5.20.198	chart.singleAxis.axisPointer.handle.throttle.....	2357
5.20.199	chart.singleAxis.axisPointer.handle.shadowBlur.....	2357
5.20.200	chart.singleAxis.axisPointer.handle.shadowColor.....	2358
5.20.201	chart.singleAxis.axisPointer.handle.shadowOffsetX.....	2358

5.20.202	chart.singleAxis.axisPointer.handle.shadowOffsetY.....	2358
5.20.203	chart.singleAxis.tooltip.object .....	2359
5.20.204	chart.singleAxis.tooltip.show .....	2359
5.20.205	chart.singleAxis.tooltip.trigger.....	2360
	Erlaubte Werte .....	2360
5.20.206	chart.singleAxis.tooltip.axisPointer.object .....	2360
5.20.207	chart.singleAxis.tooltip.axisPointer.type.....	2361
	Erlaubte Werte .....	2361
5.20.208	chart.singleAxis.tooltip.axisPointer.axis .....	2361
	Erlaubte Werte .....	2362
5.20.209	chart.singleAxis.tooltip.axisPointer.snap .....	2362
5.20.210	chart.singleAxis.tooltip.axisPointer.z .....	2362
5.20.211	chart.singleAxis.tooltip.axisPointer.label.object .....	2363
5.20.212	chart.singleAxis.tooltip.axisPointer.label.show .....	2363
5.20.213	chart.singleAxis.tooltip.axisPointer.label.precision .....	2363
5.20.214	chart.singleAxis.tooltip.axisPointer.label.formatter.....	2364
5.20.215	chart.singleAxis.tooltip.axisPointer.label.margin.....	2365
5.20.216	chart.singleAxis.tooltip.axisPointer.label.color .....	2366
5.20.217	chart.singleAxis.tooltip.axisPointer.label.fontStyle .....	2366
	Erlaubte Werte .....	2366
5.20.218	chart.singleAxis.tooltip.axisPointer.label.fontWeight .....	2367
	Erlaubte Werte .....	2367
5.20.219	chart.singleAxis.tooltip.axisPointer.label.fontFamily.....	2367
5.20.220	chart.singleAxis.tooltip.axisPointer.label.fontSize .....	2368
5.20.221	chart.singleAxis.tooltip.axisPointer.label.lineHeight .....	2368
5.20.222	chart.singleAxis.tooltip.axisPointer.label.width .....	2369
5.20.223	chart.singleAxis.tooltip.axisPointer.label.height.....	2369
5.20.224	chart.singleAxis.tooltip.axisPointer.label.textBorderColor.....	2369
5.20.225	chart.singleAxis.tooltip.axisPointer.label.textBorderWidth.....	2370
5.20.226	chart.singleAxis.tooltip.axisPointer.label.textShadowColor .....	2370
5.20.227	chart.singleAxis.tooltip.axisPointer.label.textShadowBlur.....	2371
5.20.228	chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetX .....	2371
5.20.229	chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetY.....	2371
5.20.230	chart.singleAxis.tooltip.axisPointer.label.padding.....	2372
5.20.231	chart.singleAxis.tooltip.axisPointer.label.backgroundColor .....	2372
5.20.232	chart.singleAxis.tooltip.axisPointer.label.borderColor .....	2373



5.20.233	chart.singleAxis.tooltip.axisPointer.label.borderWidth .....	2373
5.20.234	chart.singleAxis.tooltip.axisPointer.label.shadowBlur.....	2373
5.20.235	chart.singleAxis.tooltip.axisPointer.label.shadowColor .....	2374
5.20.236	chart.singleAxis.tooltip.axisPointer.label.shadowOffsetX .....	2374
5.20.237	chart.singleAxis.tooltip.axisPointer.label.shadowOffsetY.....	2375
5.20.238	chart.singleAxis.tooltip.axisPointer.lineStyle.object.....	2375
5.20.239	chart.singleAxis.tooltip.axisPointer.lineStyle.color.....	2375
5.20.240	chart.singleAxis.tooltip.axisPointer.lineStyle.width.....	2377
5.20.241	chart.singleAxis.tooltip.axisPointer.lineStyle.type .....	2377
	Erlaubte Werte .....	2377
5.20.242	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowBlur .....	2377
5.20.243	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowColor .....	2378
5.20.244	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetX .....	2378
5.20.245	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetY .....	2379
5.20.246	chart.singleAxis.tooltip.axisPointer.lineStyle.opacity .....	2379
5.20.247	chart.singleAxis.tooltip.axisPointer.shadowStyle.object.....	2379
5.20.248	chart.singleAxis.tooltip.axisPointer.shadowStyle.color.....	2380
5.20.249	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowBlur .....	2381
5.20.250	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowColor .....	2382
5.20.251	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetX .....	2382
5.20.252	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	2383
5.20.253	chart.singleAxis.tooltip.axisPointer.shadowStyle.opacity .....	2383
5.20.254	chart.singleAxis.tooltip.axisPointer.crossStyle.object .....	2383
5.20.255	chart.singleAxis.tooltip.axisPointer.crossStyle.color .....	2384
5.20.256	chart.singleAxis.tooltip.axisPointer.crossStyle.width .....	2385
5.20.257	chart.singleAxis.tooltip.axisPointer.crossStyle.type .....	2386
	Erlaubte Werte .....	2386
5.20.258	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowBlur.....	2386
5.20.259	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowColor.....	2387
5.20.260	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetX.....	2387
5.20.261	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetY.....	2387
5.20.262	chart.singleAxis.tooltip.axisPointer.crossStyle.opacity .....	2388
5.20.263	chart.singleAxis.tooltip.position .....	2388
5.20.264	chart.singleAxis.tooltip.formatter.....	2390
5.20.265	chart.singleAxis.tooltip.backgroundColor.....	2392
5.20.266	chart.singleAxis.tooltip.borderColor .....	2392

5.20.267	chart.singleAxis.tooltip.borderWidth.....	2393
5.20.268	chart.singleAxis.tooltip.padding.....	2393
5.20.269	chart.singleAxis.tooltip.textStyle.object .....	2394
5.20.270	chart.singleAxis.tooltip.textStyle.color .....	2394
5.20.271	chart.singleAxis.tooltip.textStyle.fontStyle.....	2394
	Erlaubte Werte .....	2395
5.20.272	chart.singleAxis.tooltip.textStyle.fontWeight .....	2395
	Erlaubte Werte .....	2395
5.20.273	chart.singleAxis.tooltip.textStyle.fontFamily.....	2395
5.20.274	chart.singleAxis.tooltip.textStyle.fontSize .....	2396
5.20.275	chart.singleAxis.tooltip.textStyle.lineHeight.....	2396
5.20.276	chart.singleAxis.tooltip.textStyle.width .....	2397
5.20.277	chart.singleAxis.tooltip.textStyle.height .....	2397
5.20.278	chart.singleAxis.tooltip.textStyle.textBorderColor.....	2398
5.20.279	chart.singleAxis.tooltip.textStyle.textBorderWidth.....	2398
5.20.280	chart.singleAxis.tooltip.textStyle.textShadowColor.....	2398
5.20.281	chart.singleAxis.tooltip.textStyle.textShadowBlur.....	2399
5.20.282	chart.singleAxis.tooltip.textStyle.textShadowOffsetX.....	2399
5.20.283	chart.singleAxis.tooltip.textStyle.textShadowOffsetY.....	2399
5.20.284	chart.singleAxis.tooltip.extraCssText .....	2400
5.21	ECharts (4) Text Style.....	2400
5.21.1	chart.textStyle.object .....	2400
5.21.2	chart.textStyle.color .....	2401
5.21.3	chart.textStyle.fontStyle.....	2401
	Erlaubte Werte .....	2401
5.21.4	chart.textStyle.fontWeight .....	2401
	Erlaubte Werte .....	2402
5.21.5	chart.textStyle.fontFamily.....	2402
5.21.6	chart.textStyle.fontSize .....	2402
5.21.7	chart.textStyle.lineHeight .....	2403
5.21.8	chart.textStyle.width .....	2403
5.21.9	chart.textStyle.height .....	2404
5.21.10	chart.textStyle.textBorderColor.....	2404
5.21.11	chart.textStyle.textBorderWidth.....	2405
5.21.12	chart.textStyle.textShadowColor.....	2405
5.21.13	chart.textStyle.textShadowBlur.....	2405

5.21.14	chart.textStyle.textShadowOffsetX.....	2406
5.21.15	chart.textStyle.textShadowOffsetY.....	2406
5.22	ECharts (4) Timeline.....	2407
5.22.1	chart.timeline.object .....	2407
5.22.2	chart.timeline.show .....	2409
5.22.3	chart.timeline.type .....	2410
5.22.4	chart.timeline.axisType .....	2410
	Erlaubte Werte .....	2410
5.22.5	chart.timeline.currentIndex .....	2411
5.22.6	chart.timeline.autoPlay .....	2411
5.22.7	chart.timeline.rewind .....	2411
5.22.8	chart.timeline.loop .....	2412
5.22.9	chart.timeline.playInterval.....	2412
5.22.10	chart.timeline.realtime.....	2412
5.22.11	chart.timeline.controlPosition.....	2413
	Erlaubte Werte .....	2413
5.22.12	chart.timeline.zlevel .....	2413
5.22.13	chart.timeline.z .....	2414
5.22.14	chart.timeline.left .....	2414
	Erlaubte Werte .....	2415
5.22.15	chart.timeline.top .....	2415
	Erlaubte Werte .....	2415
5.22.16	chart.timeline.right.....	2415
	Erlaubte Werte .....	2416
5.22.17	chart.timeline.bottom .....	2416
	Erlaubte Werte .....	2416
5.22.18	chart.timeline.padding.....	2416
5.22.19	chart.timeline.orient.....	2417
	Erlaubte Werte .....	2417
5.22.20	chart.timeline.inverse.....	2418
5.22.21	chart.timeline.symbol.....	2418
	Erlaubte Werte .....	2418
5.22.22	chart.timeline.symbolSize.....	2419
5.22.23	chart.timeline.symbolRotate .....	2419
5.22.24	chart.timeline.symbolKeepAspect.....	2419
5.22.25	chart.timeline.symbolOffset .....	2420

5.22.26	chart.timeline.lineStyle.object.....	2420
5.22.27	chart.timeline.lineStyle.show .....	2421
5.22.28	chart.timeline.lineStyle.color.....	2421
5.22.29	chart.timeline.lineStyle.width.....	2422
5.22.30	chart.timeline.lineStyle.type.....	2423
	Erlaubte Werte .....	2423
5.22.31	chart.timeline.lineStyle.shadowBlur .....	2423
5.22.32	chart.timeline.lineStyle.shadowColor .....	2424
5.22.33	chart.timeline.lineStyle.shadowOffsetX .....	2424
5.22.34	chart.timeline.lineStyle.shadowOffsetY .....	2424
5.22.35	chart.timeline.lineStyle.opacity.....	2425
5.22.36	chart.timeline.label.object.....	2425
5.22.37	chart.timeline.label.position.....	2425
	Erlaubte Werte .....	2426
5.22.38	chart.timeline.label.show.....	2426
5.22.39	chart.timeline.label.interval.....	2427
5.22.40	chart.timeline.label.rotate .....	2427
5.22.41	chart.timeline.label.formatter .....	2427
5.22.42	chart.timeline.label.color.....	2428
5.22.43	chart.timeline.label.fontStyle .....	2428
	Erlaubte Werte .....	2429
5.22.44	chart.timeline.label.fontWeight.....	2429
	Erlaubte Werte .....	2429
5.22.45	chart.timeline.label.fontFamily .....	2429
5.22.46	chart.timeline.label.fontSize.....	2430
5.22.47	chart.timeline.label.align .....	2430
	Erlaubte Werte .....	2431
5.22.48	chart.timeline.label.verticalAlign.....	2431
	Erlaubte Werte .....	2431
5.22.49	chart.timeline.label.lineHeight .....	2432
5.22.50	chart.timeline.label.backgroundColor .....	2432
5.22.51	chart.timeline.label.borderColor .....	2433
5.22.52	chart.timeline.label.borderWidth .....	2433
5.22.53	chart.timeline.label.borderRadius.....	2434
5.22.54	chart.timeline.label.padding.....	2434
5.22.55	chart.timeline.label.shadowColor .....	2434

5.22.56	chart.timeline.label.shadowBlur .....	2435
5.22.57	chart.timeline.label.shadowOffsetX .....	2435
5.22.58	chart.timeline.label.shadowOffsetY .....	2435
5.22.59	chart.timeline.label.width .....	2436
5.22.60	chart.timeline.label.height .....	2436
5.22.61	chart.timeline.label.textBorderColor .....	2437
5.22.62	chart.timeline.label.textBorderWidth .....	2437
5.22.63	chart.timeline.label.textShadowColor .....	2437
5.22.64	chart.timeline.label.textShadowBlur .....	2438
5.22.65	chart.timeline.label.textShadowOffsetX .....	2438
5.22.66	chart.timeline.label.textShadowOffsetY .....	2439
5.22.67	chart.timeline.label.rich.object .....	2439
5.22.68	chart.timeline.label.emphasis.object .....	2440
5.22.69	chart.timeline.label.emphasis.show .....	2441
5.22.70	chart.timeline.label.emphasis.interval .....	2441
5.22.71	chart.timeline.label.emphasis.rotate .....	2441
5.22.72	chart.timeline.label.emphasis.formatter .....	2442
5.22.73	chart.timeline.label.emphasis.color .....	2442
5.22.74	chart.timeline.label.emphasis.fontStyle .....	2443
	Erlaubte Werte .....	2443
5.22.75	chart.timeline.label.emphasis.fontWeight .....	2443
	Erlaubte Werte .....	2444
5.22.76	chart.timeline.label.emphasis.fontFamily .....	2444
5.22.77	chart.timeline.label.emphasis.fontSize .....	2444
5.22.78	chart.timeline.label.emphasis.align .....	2445
	Erlaubte Werte .....	2445
5.22.79	chart.timeline.label.emphasis.verticalAlign .....	2445
	Erlaubte Werte .....	2446
5.22.80	chart.timeline.label.emphasis.lineHeight .....	2446
5.22.81	chart.timeline.label.emphasis.backgroundColor .....	2446
5.22.82	chart.timeline.label.emphasis.borderColor .....	2447
5.22.83	chart.timeline.label.emphasis.borderWidth .....	2448
5.22.84	chart.timeline.label.emphasis.borderRadius .....	2448
5.22.85	chart.timeline.label.emphasis.padding .....	2448
5.22.86	chart.timeline.label.emphasis.shadowColor .....	2449
5.22.87	chart.timeline.label.emphasis.shadowBlur .....	2449

5.22.88	chart.timeline.label.emphasis.shadowOffsetX .....	2449
5.22.89	chart.timeline.label.emphasis.shadowOffsetY .....	2450
5.22.90	chart.timeline.label.emphasis.width .....	2450
5.22.91	chart.timeline.label.emphasis.height .....	2451
5.22.92	chart.timeline.label.emphasis.textBorderColor .....	2451
5.22.93	chart.timeline.label.emphasis.textBorderWidth .....	2451
5.22.94	chart.timeline.label.emphasis.textShadowColor .....	2452
5.22.95	chart.timeline.label.emphasis.textShadowBlur .....	2452
5.22.96	chart.timeline.label.emphasis.textShadowOffsetX .....	2453
5.22.97	chart.timeline.label.emphasis.textShadowOffsetY .....	2453
5.22.98	chart.timeline.label.emphasis.rich.object .....	2453
5.22.99	chart.timeline.itemStyle.object .....	2454
5.22.100	chart.timeline.itemStyle.color .....	2455
5.22.101	chart.timeline.itemStyle.borderColor .....	2456
5.22.102	chart.timeline.itemStyle.borderWidth .....	2457
5.22.103	chart.timeline.itemStyle.borderType .....	2457
	Erlaubte Werte .....	2457
5.22.104	chart.timeline.itemStyle.shadowBlur .....	2458
5.22.105	chart.timeline.itemStyle.shadowColor .....	2458
5.22.106	chart.timeline.itemStyle.shadowOffsetX .....	2458
5.22.107	chart.timeline.itemStyle.shadowOffsetY .....	2459
5.22.108	chart.timeline.itemStyle.opacity .....	2459
5.22.109	chart.timeline.itemStyle.emphasis.object .....	2459
5.22.110	chart.timeline.itemStyle.emphasis.color .....	2460
5.22.111	chart.timeline.itemStyle.emphasis.borderColor .....	2461
5.22.112	chart.timeline.itemStyle.emphasis.borderWidth .....	2462
5.22.113	chart.timeline.itemStyle.emphasis.borderType .....	2462
	Erlaubte Werte .....	2462
5.22.114	chart.timeline.itemStyle.emphasis.shadowBlur .....	2463
5.22.115	chart.timeline.itemStyle.emphasis.shadowColor .....	2463
5.22.116	chart.timeline.itemStyle.emphasis.shadowOffsetX .....	2463
5.22.117	chart.timeline.itemStyle.emphasis.shadowOffsetY .....	2464
5.22.118	chart.timeline.itemStyle.emphasis.opacity .....	2464
5.22.119	chart.timeline.checkpointStyle.object .....	2464
5.22.120	chart.timeline.checkpointStyle.symbol .....	2465
	Erlaubte Werte .....	2465

5.22.121	chart.timeline.checkpointStyle.symbolSize.....	2466
5.22.122	chart.timeline.checkpointStyle.symbolRotate .....	2466
5.22.123	chart.timeline.checkpointStyle.symbolKeepAspect.....	2466
5.22.124	chart.timeline.checkpointStyle.symbolOffset .....	2467
5.22.125	chart.timeline.checkpointStyle.color .....	2467
5.22.126	chart.timeline.checkpointStyle.borderWidth .....	2467
5.22.127	chart.timeline.checkpointStyle.borderColor .....	2468
5.22.128	chart.timeline.checkpointStyle.animation .....	2468
5.22.129	chart.timeline.checkpointStyle.animationDuration.....	2469
5.22.130	chart.timeline.checkpointStyle.animationEasing .....	2469
	Erlaubte Werte .....	2469
5.22.131	chart.timeline.controlStyle.object.....	2470
5.22.132	chart.timeline.controlStyle.show .....	2470
5.22.133	chart.timeline.controlStyle.showPlayBtn .....	2471
5.22.134	chart.timeline.controlStyle.showPrevBtn .....	2471
5.22.135	chart.timeline.controlStyle.showNextBtn.....	2471
5.22.136	chart.timeline.controlStyle.itemSize.....	2472
5.22.137	chart.timeline.controlStyle.itemGap .....	2472
5.22.138	chart.timeline.controlStyle.position .....	2472
	Erlaubte Werte .....	2473
5.22.139	chart.timeline.controlStyle.playIcon .....	2473
5.22.140	chart.timeline.controlStyle.stopIcon.....	2473
5.22.141	chart.timeline.controlStyle.prevIcon.....	2474
5.22.142	chart.timeline.controlStyle.nextIcon.....	2474
5.22.143	chart.timeline.controlStyle.color.....	2475
5.22.144	chart.timeline.controlStyle.borderColor.....	2475
5.22.145	chart.timeline.controlStyle.borderWidth .....	2476
5.22.146	chart.timeline.controlStyle.emphasis.object.....	2476
5.22.147	chart.timeline.controlStyle.emphasis.color.....	2476
5.22.148	chart.timeline.controlStyle.emphasis.borderColor.....	2477
5.22.149	chart.timeline.controlStyle.emphasis.borderWidth.....	2477
5.23	ECharts (4) Title .....	2477
5.23.1	chart.title.object .....	2477
5.23.2	chart.title.id .....	2478
5.23.3	chart.title.show .....	2478
5.23.4	chart.title.text .....	2479

5.23.5	chart.title.link.....	2479
5.23.6	chart.title.target.....	2479
5.23.7	chart.title.textStyle.object.....	2480
5.23.8	chart.title.textStyle.color .....	2480
5.23.9	chart.title.textStyle.fontStyle .....	2480
	Erlaubte Werte .....	2481
5.23.10	chart.title.textStyle.fontWeight .....	2481
5.23.11	chart.title.textStyle.fontFamily .....	2481
5.23.12	chart.title.textStyle.fontSize .....	2482
5.23.13	chart.title.textStyle.align .....	2482
5.23.14	chart.title.textStyle.verticalAlign .....	2483
5.23.15	chart.title.textStyle.lineHeight.....	2483
5.23.16	chart.title.textStyle.backgroundColor.....	2484
5.23.17	chart.title.textStyle.borderColor.....	2484
5.23.18	chart.title.textStyle.borderWidth.....	2485
5.23.19	chart.title.textStyle.borderRadius .....	2485
5.23.20	chart.title.textStyle.padding .....	2486
5.23.21	chart.title.textStyle.shadowColor .....	2486
5.23.22	chart.title.textStyle.shadowBlur .....	2487
5.23.23	chart.title.textStyle.shadowOffsetX.....	2487
5.23.24	chart.title.textStyle.shadowOffsetY .....	2487
5.23.25	chart.title.textStyle.width .....	2488
	Erlaubte Werte .....	2488
5.23.26	chart.title.textStyle.height .....	2488
5.23.27	chart.title.textStyle.textBorderColor .....	2489
5.23.28	chart.title.textStyle.textBorderWidth .....	2489
5.23.29	chart.title.textStyle.textShadowColor .....	2489
5.23.30	chart.title.textStyle.textShadowBlur .....	2490
5.23.31	chart.title.textStyle.textShadowOffsetX.....	2490
5.23.32	chart.title.textStyle.textShadowOffsetY .....	2490
5.23.33	chart.title.textStyle.rich.....	2491
5.23.34	chart.title.subtext .....	2492
5.23.35	chart.title.sublink.....	2493
5.23.36	chart.title.subtarget.....	2493
5.23.37	chart.title.subtextStyle.object .....	2493
5.23.38	chart.title.subtextStyle.color .....	2494



5.23.39	chart.title.subtextStyle.fontStyle.....	2494
	Erlaubte Werte .....	2494
5.23.40	chart.title.subtextStyle.fontWeight .....	2495
5.23.41	chart.title.subtextStyle.fontFamily .....	2495
5.23.42	chart.title.subtextStyle.fontSize .....	2495
5.23.43	chart.title.subtextStyle.align.....	2496
	Erlaubte Werte .....	2496
5.23.44	chart.title.subtextStyle.verticalAlign .....	2497
5.23.45	chart.title.subtextStyle.lineHeight.....	2497
5.23.46	chart.title.subtextStyle.backgroundColor.....	2498
5.23.47	chart.title.subtextStyle.borderColor.....	2498
5.23.48	chart.title.subtextStyle.borderWidth.....	2499
5.23.49	chart.title.subtextStyle.borderRadius .....	2499
5.23.50	chart.title.subtextStyle.padding .....	2499
5.23.51	chart.title.subtextStyle.shadowColor.....	2500
5.23.52	chart.title.subtextStyle.shadowBlur .....	2500
5.23.53	chart.title.subtextStyle.shadowOffsetX.....	2501
5.23.54	chart.title.subtextStyle.shadowOffsetY .....	2501
5.23.55	chart.title.subtextStyle.width .....	2502
	Erlaubte Werte .....	2502
5.23.56	chart.title.subtextStyle.height .....	2502
5.23.57	chart.title.subtextStyle.textBorderColor .....	2503
5.23.58	chart.title.subtextStyle.textBorderWidth .....	2503
5.23.59	chart.title.subtextStyle.textShadowColor.....	2504
5.23.60	chart.title.subtextStyle.textShadowBlur .....	2504
5.23.61	chart.title.subtextStyle.textShadowOffsetX.....	2504
5.23.62	chart.title.subtextStyle.textShadowOffsetY .....	2505
5.23.63	chart.title.subtextStyle.rich.....	2505
5.23.64	chart.title.triggerEvent .....	2506
5.23.65	chart.title.padding .....	2507
5.23.66	chart.title.itemGap.....	2507
5.23.67	chart.title.zlevel .....	2508
5.23.68	chart.title.z .....	2508
5.23.69	chart.title.left .....	2508
	Erlaubte Werte .....	2509
5.23.70	chart.title.top .....	2509

	Erlaubte Werte .....	2510
5.23.71	chart.title.right .....	2510
	Erlaubte Werte .....	2510
5.23.72	chart.title.bottom .....	2510
	Erlaubte Werte .....	2511
5.23.73	chart.title.backgroundColor.....	2511
5.23.74	chart.title.borderColor.....	2511
5.23.75	chart.title.borderWidth.....	2512
5.23.76	chart.title.borderRadius .....	2512
5.23.77	chart.title.shadowBlur.....	2512
5.23.78	chart.title.shadowColor.....	2513
5.23.79	chart.title.shadowOffsetX.....	2513
5.23.80	chart.title.shadowOffsetY.....	2514
5.24	<b>ECharts (4) Toolbox.....</b>	<b>2514</b>
5.24.1	chart.toolbox.id.....	2514
5.24.2	chart.toolbox.show .....	2515
5.24.3	chart.toolbox.orient.....	2515
	Erlaubte Werte .....	2515
5.24.4	chart.toolbox.itemSize .....	2515
5.24.5	chart.toolbox.itemGap .....	2516
5.24.6	chart.toolbox.showTitle .....	2516
5.24.7	chart.toolbox.feature.saveAsImage.....	2516
5.24.8	chart.toolbox.feature.saveAsImage.type .....	2517
5.24.9	chart.toolbox.feature.saveAsImage.name .....	2517
5.24.10	chart.toolbox.feature.saveAsImage.backgroundColor .....	2517
5.24.11	chart.toolbox.feature.saveAsImage.excludeComponents .....	2518
5.24.12	chart.toolbox.feature.saveAsImage.show .....	2518
5.24.13	chart.toolbox.feature.saveAsImage.title .....	2519
5.24.14	chart.toolbox.feature.saveAsImage.icon.....	2519
5.24.15	chart.toolbox.feature.saveAsImage.iconStyle .....	2519
5.24.16	chart.toolbox.feature.saveAsImage.iconStyle.color.....	2520
5.24.17	chart.toolbox.feature.saveAsImage.iconStyle.borderColor.....	2520
5.24.18	chart.toolbox.feature.saveAsImage.iconStyle.borderWidth.....	2520
5.24.19	chart.toolbox.feature.saveAsImage.iconStyle.borderType.....	2521
	Erlaubte Werte .....	2521
5.24.20	chart.toolbox.feature.saveAsImage.iconStyle.shadowBlur .....	2521

5.24.21	chart.toolbox.feature.saveAsImage.iconStyle.shadowColor .....	2522
5.24.22	chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetX .....	2522
5.24.23	chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetY .....	2522
5.24.24	chart.toolbox.feature.saveAsImage.iconStyle.opacity .....	2523
5.24.25	chart.toolbox.feature.saveAsImage.iconStyle.textPosition .....	2523
	Erlaubte Werte .....	2523
5.24.26	chart.toolbox.feature.saveAsImage.iconStyle.textAlign .....	2523
	Erlaubte Werte .....	2524
5.24.27	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.color .....	2524
5.24.28	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderColor .....	2524
5.24.29	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderWidth .....	2525
5.24.30	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderType .....	2525
	Erlaubte Werte .....	2525
5.24.31	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowBlur .....	2525
5.24.32	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowColor .....	2526
5.24.33	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetX .....	2526
5.24.34	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetY .....	2527
5.24.35	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.opacity .....	2527
5.24.36	chart.toolbox.feature.saveAsImage.pixelRatio .....	2527
5.24.37	chart.toolbox.feature.restore .....	2528
5.24.38	chart.toolbox.feature.restore.show .....	2528
5.24.39	chart.toolbox.feature.restore.title .....	2528
5.24.40	chart.toolbox.feature.restore.icon .....	2529
5.24.41	chart.toolbox.feature.restore.iconStyle .....	2529
5.24.42	chart.toolbox.feature.restore.iconStyle.color .....	2529
5.24.43	chart.toolbox.feature.restore.iconStyle.borderColor .....	2530
5.24.44	chart.toolbox.feature.restore.iconStyle.borderWidth .....	2530
5.24.45	chart.toolbox.feature.restore.iconStyle.borderType .....	2531
	Erlaubte Werte .....	2531
5.24.46	chart.toolbox.feature.restore.iconStyle.shadowBlur .....	2531
5.24.47	chart.toolbox.feature.restore.iconStyle.shadowColor .....	2531
5.24.48	chart.toolbox.feature.restore.iconStyle.shadowOffsetX .....	2532
5.24.49	chart.toolbox.feature.restore.iconStyle.shadowOffsetY .....	2532
5.24.50	chart.toolbox.feature.restore.iconStyle.opacity .....	2532
5.24.51	chart.toolbox.feature.restore.iconStyle.textPosition .....	2533
	Erlaubte Werte .....	2533

5.24.52	chart.toolbox.feature.restore.iconStyle.textAlign .....	2533
	Erlaubte Werte .....	2534
5.24.53	chart.toolbox.feature.restore.iconStyle.emphasis.color .....	2534
5.24.54	chart.toolbox.feature.restore.iconStyle.emphasis.borderColor.....	2534
5.24.55	chart.toolbox.feature.restore.iconStyle.emphasis.borderWidth.....	2534
5.24.56	chart.toolbox.feature.restore.iconStyle.emphasis.borderType .....	2535
	Erlaubte Werte .....	2535
5.24.57	chart.toolbox.feature.restore.iconStyle.emphasis.shadowBlur .....	2535
5.24.58	chart.toolbox.feature.restore.iconStyle.emphasis.shadowColor .....	2536
5.24.59	chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetX.....	2536
5.24.60	chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetY .....	2536
5.24.61	chart.toolbox.feature.restore.iconStyle.emphasis.opacity.....	2537
5.24.62	chart.toolbox.feature.dataView .....	2537
5.24.63	chart.toolbox.feature.dataView.show .....	2537
5.24.64	chart.toolbox.feature.dataView.title .....	2538
5.24.65	chart.toolbox.feature.dataView.icon .....	2538
5.24.66	chart.toolbox.feature.dataView.iconStyle.....	2538
5.24.67	chart.toolbox.feature.dataView.iconStyle.color .....	2539
5.24.68	chart.toolbox.feature.dataView.iconStyle.borderColor .....	2539
5.24.69	chart.toolbox.feature.dataView.iconStyle.borderWidth .....	2539
5.24.70	chart.toolbox.feature.dataView.iconStyle.borderType .....	2540
	Erlaubte Werte .....	2540
5.24.71	chart.toolbox.feature.dataView.iconStyle.shadowBlur .....	2540
5.24.72	chart.toolbox.feature.dataView.iconStyle.shadowColor .....	2541
5.24.73	chart.toolbox.feature.dataView.iconStyle.shadowOffsetX .....	2541
5.24.74	chart.toolbox.feature.dataView.iconStyle.shadowOffsetY .....	2541
5.24.75	chart.toolbox.feature.dataView.iconStyle.opacity .....	2542
5.24.76	chart.toolbox.feature.dataView.iconStyle.textPosition .....	2542
	Erlaubte Werte .....	2542
5.24.77	chart.toolbox.feature.dataView.iconStyle.textAlign.....	2543
	Erlaubte Werte .....	2543
5.24.78	chart.toolbox.feature.dataView.iconStyle.emphasis .....	2543
5.24.79	chart.toolbox.feature.dataView.iconStyle.emphasis.color .....	2543
5.24.80	chart.toolbox.feature.dataView.iconStyle.emphasis.borderColor .....	2544
5.24.81	chart.toolbox.feature.dataView.iconStyle.emphasis.borderWidth .....	2544
5.24.82	chart.toolbox.feature.dataView.iconStyle.emphasis.borderType.....	2544

	Erlaubte Werte .....	2545
5.24.83	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowBlur .....	2545
5.24.84	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowColor .....	2545
5.24.85	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetX .....	2546
5.24.86	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetY .....	2546
5.24.87	chart.toolbox.feature.dataView.iconStyle.emphasis.opacity .....	2546
5.24.88	chart.toolbox.feature.dataView.readOnly .....	2547
5.24.89	chart.toolbox.feature.dataView.optionToContent .....	2547
5.24.90	chart.toolbox.feature.dataView.contentToOption .....	2548
5.24.91	chart.toolbox.feature.dataView.lang .....	2548
5.24.92	chart.toolbox.feature.dataView.backgroundColor .....	2549
5.24.93	chart.toolbox.feature.dataView.textareaColor .....	2549
5.24.94	chart.toolbox.feature.dataView.textareaBorderColor .....	2549
5.24.95	chart.toolbox.feature.dataView.textColor .....	2550
5.24.96	chart.toolbox.feature.dataView.buttonColor .....	2550
5.24.97	chart.toolbox.feature.dataView.buttonTextColor .....	2550
5.24.98	chart.toolbox.feature.dataZoom .....	2551
5.24.99	chart.toolbox.feature.dataZoom.show .....	2551
5.24.100	chart.toolbox.feature.dataZoom.title .....	2552
5.24.101	chart.toolbox.feature.dataZoom.title.zoom .....	2552
5.24.102	chart.toolbox.feature.dataZoom.title.back .....	2552
5.24.103	chart.toolbox.feature.dataZoom.icon .....	2553
5.24.104	chart.toolbox.feature.dataZoom.icon.zoom .....	2553
5.24.105	chart.toolbox.feature.dataZoom.icon.back .....	2553
5.24.106	chart.toolbox.feature.dataZoom.iconStyle .....	2554
5.24.107	chart.toolbox.feature.dataZoom.iconStyle.color .....	2554
5.24.108	chart.toolbox.feature.dataZoom.iconStyle.borderColor .....	2555
5.24.109	chart.toolbox.feature.dataZoom.iconStyle.borderWidth .....	2555
5.24.110	chart.toolbox.feature.dataZoom.iconStyle.borderType .....	2555
	Erlaubte Werte .....	2556
5.24.111	chart.toolbox.feature.dataZoom.iconStyle.shadowBlur .....	2556
5.24.112	chart.toolbox.feature.dataZoom.iconStyle.shadowColor .....	2556
5.24.113	chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetX .....	2556
5.24.114	chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetY .....	2557
5.24.115	chart.toolbox.feature.dataZoom.iconStyle.opacity .....	2557
5.24.116	chart.toolbox.feature.dataZoom.iconStyle.textPosition .....	2557

	Erlaubte Werte .....	2558
5.24.117	chart.toolbox.feature.dataZoom.iconStyle.textAlign .....	2558
	Erlaubte Werte .....	2558
5.24.118	chart.toolbox.feature.dataZoom.iconStyle.emphasis.color .....	2558
5.24.119	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderColor .....	2559
5.24.120	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderWidth .....	2559
5.24.121	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderType .....	2559
	Erlaubte Werte .....	2560
5.24.122	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowBlur .....	2560
5.24.123	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowColor .....	2560
5.24.124	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetX .....	2561
5.24.125	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetY .....	2561
5.24.126	chart.toolbox.feature.dataZoom.iconStyle.emphasis.opacity .....	2561
5.24.127	chart.toolbox.feature.dataZoom.xAxisIndex .....	2562
5.24.128	chart.toolbox.feature.dataZoom.yAxisIndex .....	2562
5.24.129	chart.toolbox.feature.magicType .....	2562
5.24.130	chart.toolbox.feature.magicType.show .....	2563
5.24.131	chart.toolbox.feature.magicType.type .....	2563
5.24.132	chart.toolbox.feature.magicType.title .....	2564
5.24.133	chart.toolbox.feature.magicType.title.line .....	2564
5.24.134	chart.toolbox.feature.magicType.title.bar .....	2564
5.24.135	chart.toolbox.feature.magicType.title.stack .....	2565
5.24.136	chart.toolbox.feature.magicType.title.tiled .....	2565
5.24.137	chart.toolbox.feature.magicType.icon .....	2565
5.24.138	chart.toolbox.feature.magicType.icon.line .....	2566
5.24.139	chart.toolbox.feature.magicType.icon.bar .....	2566
5.24.140	chart.toolbox.feature.magicType.icon.stack .....	2566
5.24.141	chart.toolbox.feature.magicType.icon.tiled .....	2567
5.24.142	chart.toolbox.feature.magicType.iconStyle .....	2567
5.24.143	chart.toolbox.feature.magicType.iconStyle.color .....	2568
5.24.144	chart.toolbox.feature.magicType.iconStyle.borderColor .....	2568
5.24.145	chart.toolbox.feature.magicType.iconStyle.borderWidth .....	2568
5.24.146	chart.toolbox.feature.magicType.iconStyle.borderType .....	2569
	Erlaubte Werte .....	2569
5.24.147	chart.toolbox.feature.magicType.iconStyle.shadowBlur .....	2569
5.24.148	chart.toolbox.feature.magicType.iconStyle.shadowColor .....	2570

5.24.149	chart.toolbox.feature.magicType.iconStyle.shadowOffsetX .....	2570
5.24.150	chart.toolbox.feature.magicType.iconStyle.shadowOffsetY.....	2570
5.24.151	chart.toolbox.feature.magicType.iconStyle.opacity .....	2571
5.24.152	chart.toolbox.feature.magicType.iconStyle.textPosition .....	2571
	Erlaubte Werte .....	2571
5.24.153	chart.toolbox.feature.magicType.iconStyle.textAlign.....	2572
	Erlaubte Werte .....	2572
5.24.154	chart.toolbox.feature.magicType.iconStyle.emphasis.....	2572
5.24.155	chart.toolbox.feature.magicType.iconStyle.emphasis.color .....	2572
5.24.156	chart.toolbox.feature.magicType.iconStyle.emphasis.borderColor .....	2573
5.24.157	chart.toolbox.feature.magicType.iconStyle.emphasis.borderWidth .....	2573
5.24.158	chart.toolbox.feature.magicType.iconStyle.emphasis.borderType .....	2573
	Erlaubte Werte .....	2574
5.24.159	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowBlur .....	2574
5.24.160	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowColor .....	2574
5.24.161	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetX .....	2575
5.24.162	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetY .....	2575
5.24.163	chart.toolbox.feature.magicType.iconStyle.emphasis.opacity .....	2575
5.24.164	chart.toolbox.feature.magicType.option .....	2576
5.24.165	chart.toolbox.feature.magicType.option.line.....	2576
5.24.166	chart.toolbox.feature.magicType.option.bar .....	2576
5.24.167	chart.toolbox.feature.magicType.option.stack .....	2577
5.24.168	chart.toolbox.feature.magicType.option.tiled.....	2577
5.24.169	chart.toolbox.feature.magicType.seriesIndex .....	2577
5.24.170	chart.toolbox.feature.magicType.seriesIndex.line .....	2578
5.24.171	chart.toolbox.feature.magicType.seriesIndex.bar.....	2578
5.24.172	chart.toolbox.feature.magicType.seriesIndex.stack .....	2578
5.24.173	chart.toolbox.feature.magicType.seriesIndex.tiled.....	2579
5.24.174	chart.toolbox.feature.brush .....	2579
5.24.175	chart.toolbox.feature.brush.show .....	2579
5.24.176	chart.toolbox.feature.brush.type.....	2580
5.24.177	chart.toolbox.feature.brush.icon.....	2580
5.24.178	chart.toolbox.feature.brush.icon.rect .....	2580
5.24.179	chart.toolbox.feature.brush.icon.polygon .....	2581
5.24.180	chart.toolbox.feature.brush.icon.lineX.....	2581
5.24.181	chart.toolbox.feature.brush.icon.lineY.....	2582

5.24.182	chart.toolbox.feature.brush.icon.keep.....	2582
5.24.183	chart.toolbox.feature.brush.icon.clear.....	2583
5.24.184	chart.toolbox.feature.brush.title .....	2583
5.24.185	chart.toolbox.feature.brush.title.rect.....	2583
5.24.186	chart.toolbox.feature.brush.title.polygon.....	2584
5.24.187	chart.toolbox.feature.brush.title.lineX .....	2584
5.24.188	chart.toolbox.feature.brush.title.lineY .....	2584
5.24.189	chart.toolbox.feature.brush.title.keep .....	2585
5.24.190	chart.toolbox.feature.brush.title.clear .....	2585
5.24.191	chart.toolbox.iconStyle .....	2585
5.24.192	chart.toolbox.iconStyle.color.....	2586
5.24.193	chart.toolbox.iconStyle.borderColor.....	2586
5.24.194	chart.toolbox.iconStyle.borderWidth.....	2586
5.24.195	chart.toolbox.iconStyle.borderType.....	2587
	Erlaubte Werte .....	2587
5.24.196	chart.toolbox.iconStyle.shadowBlur .....	2587
5.24.197	chart.toolbox.iconStyle.shadowColor .....	2588
5.24.198	chart.toolbox.iconStyle.shadowOffsetX.....	2588
5.24.199	chart.toolbox.iconStyle.shadowOffsetY .....	2588
5.24.200	chart.toolbox.iconStyle.opacity.....	2589
5.24.201	chart.toolbox.iconStyle.textPosition .....	2589
	Erlaubte Werte .....	2589
5.24.202	chart.toolbox.iconStyle.textAlign .....	2590
	Erlaubte Werte .....	2590
5.24.203	chart.toolbox.iconStyle.emphasis .....	2590
5.24.204	chart.toolbox.iconStyle.emphasis.color .....	2590
5.24.205	chart.toolbox.iconStyle.emphasis.borderColor.....	2591
5.24.206	chart.toolbox.iconStyle.emphasis.borderWidth.....	2591
5.24.207	chart.toolbox.iconStyle.emphasis.borderType .....	2591
	Erlaubte Werte .....	2592
5.24.208	chart.toolbox.iconStyle.emphasis.shadowBlur .....	2592
5.24.209	chart.toolbox.iconStyle.emphasis.shadowColor .....	2592
5.24.210	chart.toolbox.iconStyle.emphasis.shadowOffsetX.....	2593
5.24.211	chart.toolbox.iconStyle.emphasis.shadowOffsetY .....	2593
5.24.212	chart.toolbox.iconStyle.emphasis.opacity.....	2593
5.24.213	chart.toolbox.zlevel .....	2594



5.24.214	chart.toolbox.z .....	2594
5.24.215	chart.toolbox.left .....	2595
	Erlaubte Werte .....	2595
5.24.216	chart.toolbox.top .....	2595
	Erlaubte Werte .....	2596
5.24.217	chart.toolbox.right .....	2596
5.24.218	chart.toolbox.bottom .....	2596
5.24.219	chart.toolbox.width .....	2597
5.24.220	chart.toolbox.height .....	2597
5.24.221	chart.toolbox.defaulticons .....	2597
5.24.222	chart.toolbox.feature.myToolToggleSeriesVisibility.show .....	2598
5.24.223	chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.show .....	2598
5.24.224	chart.toolbox.feature.RescaleYAxis.function .....	2599
5.24.225	chart.toolbox.feature.myToolRestore.show .....	2599
5.24.226	chart.toolbox.feature.myToolBarChart.show .....	2599
5.24.227	chart.toolbox.feature.myToolLineChart.show .....	2600
5.24.228	chart.toolbox.feature.myToolStackedChart.show .....	2600
5.24.229	chart.toolbox.feature.myToolRestore.additionalscript .....	2600
5.24.230	chart.toolbox.feature.myToolBarChart.additionalscript .....	2601
5.24.231	chart.toolbox.feature.myToolLineChart.additionalscript .....	2601
5.24.232	chart.toolbox.feature.myToolStackedChart.additionalscript .....	2602
5.24.233	chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript .....	2602
5.24.234	chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.additionalscript .....	2603
5.25	ECharts (4) Tooltip .....	2603
5.25.1	chart.tooltip.object .....	2603
5.25.2	chart.tooltip.show .....	2604
5.25.3	chart.tooltip.trigger .....	2604
	Erlaubte Werte .....	2605
5.25.4	chart.tooltip.axisPointer.object .....	2605
5.25.5	chart.tooltip.axisPointer.type .....	2605
	Erlaubte Werte .....	2606
5.25.6	chart.tooltip.axisPointer.axis .....	2606
	Erlaubte Werte .....	2606
5.25.7	chart.tooltip.axisPointer.snap .....	2606
5.25.8	chart.tooltip.axisPointer.z .....	2607
5.25.9	chart.tooltip.axisPointer.label.object .....	2607

5.25.10	chart.tooltip.axisPointer.label.show .....	2607
5.25.11	chart.tooltip.axisPointer.label.precision .....	2608
5.25.12	chart.tooltip.axisPointer.label.formatter .....	2608
5.25.13	chart.tooltip.axisPointer.label.margin .....	2610
5.25.14	chart.tooltip.axisPointer.label.color .....	2610
5.25.15	chart.tooltip.axisPointer.label.fontStyle .....	2610
	Erlaubte Werte .....	2611
5.25.16	chart.tooltip.axisPointer.label.fontWeight .....	2611
	Erlaubte Werte .....	2611
5.25.17	chart.tooltip.axisPointer.label.fontFamily .....	2612
5.25.18	chart.tooltip.axisPointer.label.fontSize .....	2612
5.25.19	chart.tooltip.axisPointer.label.lineHeight .....	2612
5.25.20	chart.tooltip.axisPointer.label.width .....	2613
5.25.21	chart.tooltip.axisPointer.label.height .....	2613
5.25.22	chart.tooltip.axisPointer.label.textBorderColor .....	2614
5.25.23	chart.tooltip.axisPointer.label.textBorderWidth .....	2614
5.25.24	chart.tooltip.axisPointer.label.textShadowColor .....	2615
5.25.25	chart.tooltip.axisPointer.label.textShadowBlur .....	2615
5.25.26	chart.tooltip.axisPointer.label.textShadowOffsetX .....	2615
5.25.27	chart.tooltip.axisPointer.label.textShadowOffsetY .....	2616
5.25.28	chart.tooltip.axisPointer.label.padding .....	2616
5.25.29	chart.tooltip.axisPointer.label.backgroundColor .....	2617
5.25.30	chart.tooltip.axisPointer.label.borderColor .....	2617
5.25.31	chart.tooltip.axisPointer.label.borderWidth .....	2618
5.25.32	chart.tooltip.axisPointer.label.shadowBlur .....	2618
5.25.33	chart.tooltip.axisPointer.label.shadowColor .....	2618
5.25.34	chart.tooltip.axisPointer.label.shadowOffsetX .....	2619
5.25.35	chart.tooltip.axisPointer.label.shadowOffsetY .....	2619
5.25.36	chart.tooltip.axisPointer.lineStyle.object .....	2620
5.25.37	chart.tooltip.axisPointer.lineStyle.color .....	2620
5.25.38	chart.tooltip.axisPointer.lineStyle.width .....	2621
5.25.39	chart.tooltip.axisPointer.lineStyle.type .....	2622
	Erlaubte Werte .....	2622
5.25.40	chart.tooltip.axisPointer.lineStyle.shadowBlur .....	2622
5.25.41	chart.tooltip.axisPointer.lineStyle.shadowColor .....	2623
5.25.42	chart.tooltip.axisPointer.lineStyle.shadowOffsetX .....	2623

5.25.43	chart.tooltip.axisPointer.lineStyle.shadowOffsetY .....	2623
5.25.44	chart.tooltip.axisPointer.lineStyle.opacity .....	2624
5.25.45	chart.tooltip.axisPointer.shadowStyle.object .....	2624
5.25.46	chart.tooltip.axisPointer.shadowStyle.color .....	2624
5.25.47	chart.tooltip.axisPointer.shadowStyle.shadowBlur .....	2626
5.25.48	chart.tooltip.axisPointer.shadowStyle.shadowColor .....	2627
5.25.49	chart.tooltip.axisPointer.shadowStyle.shadowOffsetX .....	2627
5.25.50	chart.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	2628
5.25.51	chart.tooltip.axisPointer.shadowStyle.opacity .....	2628
5.25.52	chart.tooltip.axisPointer.crossStyle.object .....	2628
5.25.53	chart.tooltip.axisPointer.crossStyle.color .....	2629
5.25.54	chart.tooltip.axisPointer.crossStyle.width .....	2630
5.25.55	chart.tooltip.axisPointer.crossStyle.type .....	2631
	Erlaubte Werte .....	2631
5.25.56	chart.tooltip.axisPointer.crossStyle.shadowBlur .....	2631
5.25.57	chart.tooltip.axisPointer.crossStyle.shadowColor .....	2632
5.25.58	chart.tooltip.axisPointer.crossStyle.shadowOffsetX .....	2632
5.25.59	chart.tooltip.axisPointer.crossStyle.shadowOffsetY .....	2632
5.25.60	chart.tooltip.axisPointer.crossStyle.opacity .....	2633
5.25.61	chart.tooltip.showContent .....	2633
5.25.62	chart.tooltip.alwaysShowContent .....	2633
5.25.63	chart.tooltip.triggerOn .....	2634
	Erlaubte Werte .....	2634
5.25.64	chart.tooltip.showDelay .....	2635
5.25.65	chart.tooltip.hideDelay .....	2635
5.25.66	chart.tooltip.enterable .....	2635
5.25.67	chart.tooltip.renderMode .....	2636
	Erlaubte Werte .....	2636
5.25.68	chart.tooltip.confine .....	2636
5.25.69	chart.tooltip.transitionDuration .....	2637
5.25.70	chart.tooltip.position .....	2637
5.25.71	chart.tooltip.formatter .....	2638
5.25.72	chart.tooltip.backgroundColor .....	2641
5.25.73	chart.tooltip.borderColor .....	2641
5.25.74	chart.tooltip.borderWidth .....	2642
5.25.75	chart.tooltip.padding .....	2642

5.25.76	chart.tooltip.textStyle.object .....	2643
5.25.77	chart.tooltip.textStyle.color .....	2643
5.25.78	chart.tooltip.textStyle.fontStyle .....	2643
	Erlaubte Werte .....	2644
5.25.79	chart.tooltip.textStyle.fontWeight .....	2644
	Erlaubte Werte .....	2644
5.25.80	chart.tooltip.textStyle.fontFamily .....	2644
5.25.81	chart.tooltip.textStyle.fontSize .....	2645
5.25.82	chart.tooltip.textStyle.lineHeight .....	2645
5.25.83	chart.tooltip.textStyle.width .....	2646
5.25.84	chart.tooltip.textStyle.height .....	2646
5.25.85	chart.tooltip.textStyle.textBorderColor .....	2647
5.25.86	chart.tooltip.textStyle.textBorderWidth .....	2647
5.25.87	chart.tooltip.textStyle.textShadowColor .....	2647
5.25.88	chart.tooltip.textStyle.textShadowBlur .....	2648
5.25.89	chart.tooltip.textStyle.textShadowOffsetX .....	2648
5.25.90	chart.tooltip.textStyle.textShadowOffsetY .....	2648
5.25.91	chart.tooltip.extraCssText .....	2649
5.26	<b>ECharts (4) Visual Map .....</b>	<b>2649</b>
5.26.1	chart.visualMap.object .....	2649
5.26.2	chart.visualMap.type .....	2651
	Erlaubte Werte .....	2651
5.26.3	chart.visualMap.id .....	2651
5.26.4	chart.visualMap.min .....	2652
5.26.5	chart.visualMap.max .....	2652
5.26.6	chart.visualMap.range .....	2653
5.26.7	chart.visualMap.calculable .....	2654
5.26.8	chart.visualMap.realtime .....	2654
5.26.9	chart.visualMap.inverse .....	2655
5.26.10	chart.visualMap.precision .....	2655
5.26.11	chart.visualMap.itemWidth .....	2656
5.26.12	chart.visualMap.itemHeight .....	2656
5.26.13	chart.visualMap.align .....	2657
	Erlaubte Werte .....	2657
5.26.14	chart.visualMap.text .....	2657
5.26.15	chart.visualMap.textGap .....	2658

5.26.16	chart.visualMap.show .....	2658
5.26.17	chart.visualMap.dimension .....	2658
5.26.18	chart.visualMap.seriesIndex .....	2659
5.26.19	chart.visualMap.hoverLink .....	2659
5.26.20	chart.visualMap.inRange .....	2660
5.26.21	chart.visualMap.outOfRange .....	2665
5.26.22	chart.visualMap.controller.object .....	2670
5.26.23	chart.visualMap.controller.inRange .....	2670
5.26.24	chart.visualMap.controller.outOfRange .....	2675
5.26.25	chart.visualMap.zlevel .....	2680
5.26.26	chart.visualMap.z .....	2681
5.26.27	chart.visualMap.left .....	2681
	Erlaubte Werte .....	2682
5.26.28	chart.visualMap.top .....	2682
	Erlaubte Werte .....	2682
5.26.29	chart.visualMap.right .....	2682
	Erlaubte Werte .....	2683
5.26.30	chart.visualMap.bottom .....	2683
	Erlaubte Werte .....	2683
5.26.31	chart.visualMap.orient .....	2683
	Erlaubte Werte .....	2684
5.26.32	chart.visualMap.padding .....	2684
5.26.33	chart.visualMap.backgroundColor .....	2684
5.26.34	chart.visualMap.borderColor .....	2685
5.26.35	chart.visualMap.borderWidth .....	2685
5.26.36	chart.visualMap.color .....	2686
5.26.37	chart.visualMap.textStyle.object .....	2686
5.26.38	chart.visualMap.textStyle.color .....	2686
5.26.39	chart.visualMap.textStyle.fontStyle .....	2687
	Erlaubte Werte .....	2687
5.26.40	chart.visualMap.textStyle.fontWeight .....	2687
	Erlaubte Werte .....	2688
5.26.41	chart.visualMap.textStyle.fontFamily .....	2688
5.26.42	chart.visualMap.textStyle.fontSize .....	2688
5.26.43	chart.visualMap.textStyle.lineHeight .....	2689
5.26.44	chart.visualMap.textStyle.width .....	2689

5.26.45	chart.visualMap.textStyle.height .....	2690
5.26.46	chart.visualMap.textStyle.textBorderColor .....	2690
5.26.47	chart.visualMap.textStyle.textBorderWidth .....	2690
5.26.48	chart.visualMap.textStyle.textShadowColor .....	2691
5.26.49	chart.visualMap.textStyle.textShadowBlur .....	2691
5.26.50	chart.visualMap.textStyle.textShadowOffsetX .....	2691
5.26.51	chart.visualMap.textStyle.textShadowOffsetY .....	2692
5.26.52	chart.visualMap.formatter .....	2692
5.26.53	chart.visualMap.splitNumber .....	2693
5.26.54	chart.visualMap.pieces .....	2693
5.26.55	chart.visualMap.categories .....	2694
5.26.56	chart.visualMap.minOpen .....	2695
5.26.57	chart.visualMap.maxOpen .....	2695
5.26.58	chart.visualMap.selectedMode .....	2695
	Erlaubte Werte .....	2696
5.26.59	chart.visualMap.showLabel .....	2696
<b>6</b>	<b>ECharts (5) [Options] .....</b>	<b>2697</b>
6.1	ECharts (5) Angle Axis .....	2697
6.1.1	chart.angleAxis.object .....	2697
6.1.2	chart.angleAxis.id .....	2697
6.1.3	chart.angleAxis.polarIndex .....	2698
6.1.4	chart.angleAxis.startAngle .....	2698
6.1.5	chart.angleAxis.clockwise .....	2698
6.1.6	chart.angleAxis.type .....	2699
	Erlaubte Werte .....	2699
6.1.7	chart.angleAxis.boundaryGap .....	2699
6.1.8	chart.angleAxis.min .....	2700
6.1.9	chart.angleAxis.max .....	2700
6.1.10	chart.angleAxis.scale .....	2701
6.1.11	chart.angleAxis.splitNumber .....	2701
6.1.12	chart.angleAxis.minInterval .....	2702
6.1.13	chart.angleAxis.interval .....	2702
6.1.14	chart.angleAxis.logBase .....	2703
6.1.15	chart.angleAxis.silent .....	2703
6.1.16	chart.angleAxis.triggerEvent .....	2703

6.1.17	chart.angleAxis.axisLine.object.....	2704
6.1.18	chart.angleAxis.axisLine.show .....	2704
6.1.19	chart.angleAxis.axisLine.symbol .....	2705
	Erlaubte Werte .....	2705
6.1.20	chart.angleAxis.axisLine.symbolSize .....	2705
6.1.21	chart.angleAxis.axisLine.symbolOffset.....	2706
6.1.22	chart.angleAxis.axisLine.lineStyle.object .....	2706
6.1.23	chart.angleAxis.axisLine.lineStyle.color .....	2706
6.1.24	chart.angleAxis.axisLine.lineStyle.width .....	2707
6.1.25	chart.angleAxis.axisLine.lineStyle.type .....	2708
	Erlaubte Werte .....	2708
6.1.26	chart.angleAxis.axisLine.lineStyle.shadowBlur .....	2708
6.1.27	chart.angleAxis.axisLine.lineStyle.shadowColor .....	2709
6.1.28	chart.angleAxis.axisLine.lineStyle.shadowOffsetX .....	2709
6.1.29	chart.angleAxis.axisLine.lineStyle.shadowOffsetY .....	2709
6.1.30	chart.angleAxis.axisLine.lineStyle.opacity .....	2710
6.1.31	chart.angleAxis.axisTick.object.....	2710
6.1.32	chart.angleAxis.axisTick.show .....	2711
6.1.33	chart.angleAxis.axisTick.alignWithLabel .....	2711
6.1.34	chart.angleAxis.axisTick.interval.....	2711
6.1.35	chart.angleAxis.axisTick.inside .....	2712
6.1.36	chart.angleAxis.axisTick.length.....	2712
6.1.37	chart.angleAxis.axisTick.lineStyle.object .....	2712
6.1.38	chart.angleAxis.axisTick.lineStyle.color .....	2713
6.1.39	chart.angleAxis.axisTick.lineStyle.width .....	2713
6.1.40	chart.angleAxis.axisTick.lineStyle.type .....	2713
	Erlaubte Werte .....	2714
6.1.41	chart.angleAxis.axisTick.lineStyle.shadowBlur.....	2714
6.1.42	chart.angleAxis.axisTick.lineStyle.shadowColor.....	2714
6.1.43	chart.angleAxis.axisTick.lineStyle.shadowOffsetX .....	2715
6.1.44	chart.angleAxis.axisTick.lineStyle.shadowOffsetY.....	2715
6.1.45	chart.angleAxis.axisTick.lineStyle.opacity .....	2715
6.1.46	chart.angleAxis.axisLabel.object.....	2716
6.1.47	chart.angleAxis.axisLabel.show .....	2716
6.1.48	chart.angleAxis.axisLabel.interval .....	2716
6.1.49	chart.angleAxis.axisLabel.inside .....	2717

6.1.50	chart.angleAxis.axisLabel.margin .....	2717
6.1.51	chart.angleAxis.axisLabel.formatter .....	2717
6.1.52	chart.angleAxis.axisLabel.showMinLabel .....	2718
6.1.53	chart.angleAxis.axisLabel.showMaxLabel .....	2719
6.1.54	chart.angleAxis.axisLabel.color .....	2719
6.1.55	chart.angleAxis.axisLabel.fontStyle .....	2720
	Erlaubte Werte .....	2720
6.1.56	chart.angleAxis.axisLabel.fontWeight .....	2720
	Erlaubte Werte .....	2720
6.1.57	chart.angleAxis.axisLabel.fontFamily .....	2721
6.1.58	chart.angleAxis.axisLabel.fontSize .....	2721
6.1.59	chart.angleAxis.axisLabel.align .....	2721
	Erlaubte Werte .....	2722
6.1.60	chart.angleAxis.axisLabel.verticalAlign .....	2722
	Erlaubte Werte .....	2723
6.1.61	chart.angleAxis.axisLabel.lineHeight .....	2723
6.1.62	chart.angleAxis.axisLabel.backgroundColor .....	2723
6.1.63	chart.angleAxis.axisLabel.borderColor .....	2724
6.1.64	chart.angleAxis.axisLabel.borderWidth .....	2724
6.1.65	chart.angleAxis.axisLabel.borderRadius .....	2725
6.1.66	chart.angleAxis.axisLabel.padding .....	2725
6.1.67	chart.angleAxis.axisLabel.shadowColor .....	2726
6.1.68	chart.angleAxis.axisLabel.shadowBlur .....	2726
6.1.69	chart.angleAxis.axisLabel.shadowOffsetX .....	2726
6.1.70	chart.angleAxis.axisLabel.shadowOffsetY .....	2727
6.1.71	chart.angleAxis.axisLabel.width .....	2727
6.1.72	chart.angleAxis.axisLabel.height .....	2728
6.1.73	chart.angleAxis.axisLabel.textBorderColor .....	2728
6.1.74	chart.angleAxis.axisLabel.textBorderWidth .....	2728
6.1.75	chart.angleAxis.axisLabel.textShadowColor .....	2729
6.1.76	chart.angleAxis.axisLabel.textShadowBlur .....	2729
6.1.77	chart.angleAxis.axisLabel.textShadowOffsetX .....	2729
6.1.78	chart.angleAxis.axisLabel.textShadowOffsetY .....	2730
6.1.79	chart.angleAxis.axisLabel.rich.object .....	2730
6.1.80	chart.angleAxis.splitLine.object .....	2731
6.1.81	chart.angleAxis.splitLine.show .....	2732



6.1.82	chart.angleAxis.splitLine.interval.....	2732
6.1.83	chart.angleAxis.splitLine.lineStyle.object .....	2732
6.1.84	chart.angleAxis.splitLine.lineStyle.color .....	2733
6.1.85	chart.angleAxis.splitLine.lineStyle.width .....	2733
6.1.86	chart.angleAxis.splitLine.lineStyle.type .....	2734
	Erlaubte Werte .....	2734
6.1.87	chart.angleAxis.splitLine.lineStyle.shadowBlur.....	2734
6.1.88	chart.angleAxis.splitLine.lineStyle.shadowColor.....	2735
6.1.89	chart.angleAxis.splitLine.lineStyle.shadowOffsetX.....	2735
6.1.90	chart.angleAxis.splitLine.lineStyle.shadowOffsetY.....	2735
6.1.91	chart.angleAxis.splitLine.lineStyle.opacity .....	2736
6.1.92	chart.angleAxis.splitArea.object .....	2736
6.1.93	chart.angleAxis.splitArea.interval .....	2736
6.1.94	chart.angleAxis.splitArea.show .....	2737
6.1.95	chart.angleAxis.splitArea.areaStyle.object.....	2737
6.1.96	chart.angleAxis.splitArea.areaStyle.color .....	2737
6.1.97	chart.angleAxis.splitArea.areaStyle.shadowBlur .....	2738
6.1.98	chart.angleAxis.splitArea.areaStyle.shadowColor .....	2738
6.1.99	chart.angleAxis.splitArea.areaStyle.shadowOffsetX.....	2739
6.1.100	chart.angleAxis.splitArea.areaStyle.shadowOffsetY.....	2739
6.1.101	chart.angleAxis.splitArea.areaStyle.opacity.....	2739
6.1.102	chart.angleAxis.axisPointer.object .....	2740
6.1.103	chart.angleAxis.axisPointer.show .....	2740
6.1.104	chart.angleAxis.axisPointer.type .....	2740
	Erlaubte Werte .....	2741
6.1.105	chart.angleAxis.axisPointer.snap.....	2741
6.1.106	chart.angleAxis.axisPointer.z .....	2741
6.1.107	chart.angleAxis.axisPointer.label.object .....	2742
6.1.108	chart.angleAxis.axisPointer.label.show.....	2742
6.1.109	chart.angleAxis.axisPointer.label.precision .....	2742
6.1.110	chart.angleAxis.axisPointer.label.formatter .....	2743
6.1.111	chart.angleAxis.axisPointer.label.margin.....	2744
6.1.112	chart.angleAxis.axisPointer.label.color .....	2745
6.1.113	chart.angleAxis.axisPointer.label.fontStyle .....	2745
	Erlaubte Werte .....	2745
6.1.114	chart.angleAxis.axisPointer.label.fontWeight.....	2746

Erlaubte Werte .....	2746
6.1.115 chart.angleAxis.axisPointer.label.fontFamily.....	2746
6.1.116 chart.angleAxis.axisPointer.label.fontSize .....	2747
6.1.117 chart.angleAxis.axisPointer.label.lineHeight .....	2747
6.1.118 chart.angleAxis.axisPointer.label.width .....	2748
6.1.119 chart.angleAxis.axisPointer.label.height .....	2748
6.1.120 chart.angleAxis.axisPointer.label.textBorderColor.....	2748
6.1.121 chart.angleAxis.axisPointer.label.textBorderWidth.....	2749
6.1.122 chart.angleAxis.axisPointer.label.textShadowColor .....	2749
6.1.123 chart.angleAxis.axisPointer.label.textShadowBlur.....	2750
6.1.124 chart.angleAxis.axisPointer.label.textShadowOffsetX .....	2750
6.1.125 chart.angleAxis.axisPointer.label.textShadowOffsetY.....	2750
6.1.126 chart.angleAxis.axisPointer.label.padding.....	2751
6.1.127 chart.angleAxis.axisPointer.label.backgroundColor .....	2751
6.1.128 chart.angleAxis.axisPointer.label.borderColor .....	2752
6.1.129 chart.angleAxis.axisPointer.label.borderWidth .....	2752
6.1.130 chart.angleAxis.axisPointer.label.shadowBlur.....	2752
6.1.131 chart.angleAxis.axisPointer.label.shadowColor .....	2753
6.1.132 chart.angleAxis.axisPointer.label.shadowOffsetX .....	2753
6.1.133 chart.angleAxis.axisPointer.label.shadowOffsetY.....	2754
6.1.134 chart.angleAxis.axisPointer.lineStyle.object.....	2754
6.1.135 chart.angleAxis.axisPointer.lineStyle.color .....	2754
6.1.136 chart.angleAxis.axisPointer.lineStyle.width.....	2756
6.1.137 chart.angleAxis.axisPointer.lineStyle.type .....	2756
Erlaubte Werte .....	2756
6.1.138 chart.angleAxis.axisPointer.lineStyle.shadowBlur .....	2756
6.1.139 chart.angleAxis.axisPointer.lineStyle.shadowColor .....	2757
6.1.140 chart.angleAxis.axisPointer.lineStyle.shadowOffsetX .....	2757
6.1.141 chart.angleAxis.axisPointer.lineStyle.shadowOffsetY .....	2758
6.1.142 chart.angleAxis.axisPointer.lineStyle.opacity.....	2758
6.1.143 chart.angleAxis.axisPointer.shadowStyle.object.....	2758
6.1.144 chart.angleAxis.axisPointer.shadowStyle.color .....	2759
6.1.145 chart.angleAxis.axisPointer.shadowStyle.shadowBlur .....	2760
6.1.146 chart.angleAxis.axisPointer.shadowStyle.shadowColor .....	2761
6.1.147 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetX .....	2761
6.1.148 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetY .....	2762

6.1.149	chart.angleAxis.axisPointer.shadowStyle.opacity .....	2762
6.1.150	chart.angleAxis.axisPointer.triggerTooltip .....	2762
6.1.151	chart.angleAxis.axisPointer.value .....	2763
6.1.152	chart.angleAxis.axisPointer.status .....	2763
	Erlaubte Werte .....	2763
6.1.153	chart.angleAxis.axisPointer.handle.object .....	2763
6.1.154	chart.angleAxis.axisPointer.handle.show .....	2764
6.1.155	chart.angleAxis.axisPointer.handle.icon .....	2764
6.1.156	chart.angleAxis.axisPointer.handle.size .....	2765
6.1.157	chart.angleAxis.axisPointer.handle.margin .....	2765
6.1.158	chart.angleAxis.axisPointer.handle.color .....	2765
6.1.159	chart.angleAxis.axisPointer.handle.throttle .....	2766
6.1.160	chart.angleAxis.axisPointer.handle.shadowBlur .....	2766
6.1.161	chart.angleAxis.axisPointer.handle.shadowColor .....	2767
6.1.162	chart.angleAxis.axisPointer.handle.shadowOffsetX .....	2767
6.1.163	chart.angleAxis.axisPointer.handle.shadowOffsetY .....	2767
6.1.164	chart.angleAxis.zlevel .....	2768
6.1.165	chart.angleAxis.z .....	2768
6.1.166	chart.angleAxis.axisLine.lineStyle.dashOffset .....	2769
6.1.167	chart.angleAxis.axisLine.lineStyle.cap .....	2769
	Erlaubte Werte .....	2769
6.1.168	chart.angleAxis.axisLine.lineStyle.join .....	2770
	Erlaubte Werte .....	2770
6.1.169	chart.angleAxis.axisLine.lineStyle.miterLimit .....	2770
6.1.170	chart.angleAxis.axisTick.lineStyle.dashOffset .....	2771
6.1.171	chart.angleAxis.axisTick.lineStyle.cap .....	2771
	Erlaubte Werte .....	2771
6.1.172	chart.angleAxis.axisTick.lineStyle.join .....	2772
	Erlaubte Werte .....	2772
6.1.173	chart.angleAxis.axisTick.lineStyle.miterLimit .....	2772
6.1.174	chart.angleAxis.axisLabel.borderType .....	2773
	Erlaubte Werte .....	2773
6.1.175	chart.angleAxis.axisLabel.borderDashOffset .....	2773
6.1.176	chart.angleAxis.axisLabel.overflow .....	2773
	Erlaubte Werte .....	2774
6.1.177	chart.angleAxis.axisLabel.ellipsis .....	2774

6.1.178	chart.angleAxis.axisLabel.textBorderType .....	2774
	Erlaubte Werte .....	2775
6.1.179	chart.angleAxis.axisLabel.textBorderDashOffset .....	2775
6.1.180	chart.angleAxis.splitLine.lineStyle.dashOffset .....	2775
6.1.181	chart.angleAxis.splitLine.lineStyle.cap .....	2776
	Erlaubte Werte .....	2776
6.1.182	chart.angleAxis.splitLine.lineStyle.join .....	2776
	Erlaubte Werte .....	2777
6.1.183	chart.angleAxis.splitLine.lineStyle.miterLimit .....	2777
6.1.184	chart.angleAxis.axisPointer.label.textBorderType .....	2777
	Erlaubte Werte .....	2778
6.1.185	chart.angleAxis.axisPointer.label.textBorderDashOffset .....	2778
6.1.186	chart.angleAxis.axisLabel.label.overflow .....	2778
	Erlaubte Werte .....	2779
6.1.187	chart.angleAxis.axisLabel.label.ellipsis .....	2779
6.1.188	chart.angleAxis.axisPointer.lineStyle.dashOffset .....	2779
6.1.189	chart.angleAxis.axisPointer.lineStyle.cap .....	2779
	Erlaubte Werte .....	2780
6.1.190	chart.angleAxis.axisPointer.lineStyle.join .....	2780
	Erlaubte Werte .....	2780
6.1.191	chart.angleAxis.axisPointer.lineStyle.miterLimit .....	2781
6.1.192	chart.angleAxis.minorSplitLine.object .....	2781
6.1.193	chart.angleAxis.minorSplitLine.show .....	2781
6.1.194	chart.angleAxis.minorSplitLine.lineStyle.object .....	2782
6.1.195	chart.angleAxis.minorSplitLine.lineStyle.color .....	2782
6.1.196	chart.angleAxis.minorSplitLine.lineStyle.width .....	2782
6.1.197	chart.angleAxis.minorSplitLine.lineStyle.type .....	2783
	Erlaubte Werte .....	2783
6.1.198	chart.angleAxis.minorSplitLine.lineStyle.dashOffset .....	2783
6.1.199	chart.angleAxis.minorSplitLine.lineStyle.cap .....	2784
	Erlaubte Werte .....	2784
6.1.200	chart.angleAxis.minorSplitLine.lineStyle.join .....	2784
	Erlaubte Werte .....	2785
6.1.201	chart.angleAxis.minorSplitLine.lineStyle.miterLimit .....	2785
6.1.202	chart.angleAxis.minorSplitLine.lineStyle.shadowBlur .....	2785
6.1.203	chart.angleAxis.minorSplitLine.lineStyle.shadowColor .....	2786

6.1.204	chart.angleAxis.minorSplitLine.lineStyle.shadowOffsetX .....	2786
6.1.205	chart.angleAxis.minorSplitLine.lineStyle.shadowOffsetY .....	2786
6.1.206	chart.angleAxis.minorSplitLine.lineStyle.opacity .....	2787
6.1.207	chart.angleAxis.minorTick.object .....	2787
6.1.208	chart.angleAxis.minorTick.show .....	2787
6.1.209	chart.angleAxis.minorTick.splitNumber .....	2788
6.1.210	chart.angleAxis.minorTick.length .....	2788
6.1.211	chart.angleAxis.minorTick.lineStyle.object.....	2789
6.1.212	chart.angleAxis.minorTick.lineStyle.color.....	2789
6.1.213	chart.angleAxis.minorTick.lineStyle.width.....	2789
6.1.214	chart.angleAxis.minorTick.lineStyle.type.....	2790
	Erlaubte Werte .....	2790
6.1.215	chart.angleAxis.minorTick.lineStyle.dashOffset .....	2790
6.1.216	chart.angleAxis.minorTick.lineStyle.cap .....	2791
	Erlaubte Werte .....	2791
6.1.217	chart.angleAxis.minorTick.lineStyle.join .....	2791
	Erlaubte Werte .....	2792
6.1.218	chart.angleAxis.minorTick.lineStyle.miterLimit .....	2792
6.1.219	chart.angleAxis.minorTick.lineStyle.shadowBlur .....	2792
6.1.220	chart.angleAxis.minorTick.lineStyle.shadowColor .....	2793
6.1.221	chart.angleAxis.minorTick.lineStyle.shadowOffsetX.....	2793
6.1.222	chart.angleAxis.minorTick.lineStyle.shadowOffsetY .....	2793
6.1.223	chart.angleAxis.minorTick.lineStyle.opacity.....	2794
6.1.224	chart.angleAxis.maxInterval.....	2794
6.1.225	chart.axisPointer.label.ellipsis.....	2795
6.2	<b>ECharts (5) Axis.....</b>	<b>2795</b>
6.2.1	chart.yAxis.{x}.object .....	2795
6.2.2	chart.yAxis.{x}.id.....	2795
6.2.3	chart.yAxis.{x}.show .....	2796
6.2.4	chart.yAxis.{x}.gridIndex .....	2796
6.2.5	chart.yAxis.{x}.position .....	2797
	Erlaubte Werte .....	2797
6.2.6	chart.yAxis.{x}.offset .....	2797
6.2.7	chart.yAxis.{x}.type.....	2797
	Erlaubte Werte .....	2798
6.2.8	chart.yAxis.{x}.name .....	2798

6.2.9	chart.yAxis.{x}.nameLocation.....	2798
	Erlaubte Werte .....	2799
6.2.10	chart.yAxis.{x}.nameTextStyle.object .....	2799
6.2.11	chart.yAxis.{x}.nameTextStyle.color .....	2799
6.2.12	chart.yAxis.{x}.nameTextStyle.fontStyle .....	2800
	Erlaubte Werte .....	2800
6.2.13	chart.yAxis.{x}.nameTextStyle.fontWeight .....	2800
	Erlaubte Werte .....	2800
6.2.14	chart.yAxis.{x}.nameTextStyle.fontFamily.....	2801
6.2.15	chart.yAxis.{x}.nameTextStyle.fontSize .....	2801
6.2.16	chart.yAxis.{x}.nameTextStyle.align .....	2801
	Erlaubte Werte .....	2802
6.2.17	chart.yAxis.{x}.nameTextStyle.verticalAlign.....	2802
	Erlaubte Werte .....	2803
6.2.18	chart.yAxis.{x}.nameTextStyle.lineHeight .....	2803
6.2.19	chart.yAxis.{x}.nameTextStyle.backgroundColor .....	2803
6.2.20	chart.yAxis.{x}.nameTextStyle.borderColor .....	2804
6.2.21	chart.yAxis.{x}.nameTextStyle.borderWidth .....	2804
6.2.22	chart.yAxis.{x}.nameTextStyle.borderRadius .....	2805
6.2.23	chart.yAxis.{x}.nameTextStyle.padding.....	2805
6.2.24	chart.yAxis.{x}.nameTextStyle.shadowColor .....	2806
6.2.25	chart.yAxis.{x}.nameTextStyle.shadowBlur.....	2806
6.2.26	chart.yAxis.{x}.nameTextStyle.shadowOffsetX .....	2806
6.2.27	chart.yAxis.{x}.nameTextStyle.shadowOffsetY.....	2807
6.2.28	chart.yAxis.{x}.nameTextStyle.width .....	2807
6.2.29	chart.yAxis.{x}.nameTextStyle.height .....	2808
6.2.30	chart.yAxis.{x}.nameTextStyle.textBorderColor.....	2808
6.2.31	chart.yAxis.{x}.nameTextStyle.textBorderWidth.....	2808
6.2.32	chart.yAxis.{x}.nameTextStyle.textShadowColor.....	2809
6.2.33	chart.yAxis.{x}.nameTextStyle.textShadowBlur.....	2809
6.2.34	chart.yAxis.{x}.nameTextStyle.textShadowOffsetX .....	2809
6.2.35	chart.yAxis.{x}.nameTextStyle.textShadowOffsetY.....	2810
6.2.36	chart.yAxis.{x}.nameTextStyle.rich.object.....	2810
6.2.37	chart.yAxis.{x}.nameGap.....	2811
6.2.38	chart.yAxis.{x}.nameRotate .....	2812
6.2.39	chart.yAxis.{x}.inverse.....	2812

6.2.40	chart.yAxis.{x}.boundaryGap .....	2812
6.2.41	chart.yAxis.{x}.min .....	2813
6.2.42	chart.yAxis.{x}.max .....	2813
6.2.43	chart.yAxis.{x}.scale .....	2814
6.2.44	chart.yAxis.{x}.splitNumber .....	2814
6.2.45	chart.yAxis.{x}.minInterval .....	2815
6.2.46	chart.yAxis.{x}.interval .....	2815
6.2.47	chart.yAxis.{x}.logBase .....	2816
6.2.48	chart.yAxis.{x}.silent .....	2816
6.2.49	chart.yAxis.{x}.triggerEvent .....	2816
6.2.50	chart.yAxis.{x}.axisLine.object .....	2817
6.2.51	chart.yAxis.{x}.axisLine.show .....	2817
6.2.52	chart.yAxis.{x}.axisLine.onZero .....	2818
6.2.53	chart.yAxis.{x}.axisLine.onZeroAxisIndex .....	2818
6.2.54	chart.yAxis.{x}.axisLine.symbol .....	2818
	Erlaubte Werte .....	2819
6.2.55	chart.yAxis.{x}.axisLine.symbolSize .....	2819
6.2.56	chart.yAxis.{x}.axisLine.symbolOffset .....	2819
6.2.57	chart.yAxis.{x}.axisLine.lineStyle.object .....	2820
6.2.58	chart.yAxis.{x}.axisLine.lineStyle.color .....	2820
6.2.59	chart.yAxis.{x}.axisLine.lineStyle.width .....	2821
6.2.60	chart.yAxis.{x}.axisLine.lineStyle.type .....	2822
	Erlaubte Werte .....	2822
6.2.61	chart.yAxis.{x}.axisLine.lineStyle.shadowBlur .....	2822
6.2.62	chart.yAxis.{x}.axisLine.lineStyle.shadowColor .....	2823
6.2.63	chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetX .....	2823
6.2.64	chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetY .....	2823
6.2.65	chart.yAxis.{x}.axisLine.lineStyle.opacity .....	2824
6.2.66	chart.yAxis.{x}.axisTick.object .....	2824
6.2.67	chart.yAxis.{x}.axisTick.show .....	2824
6.2.68	chart.yAxis.{x}.axisTick.alignWithLabel .....	2825
6.2.69	chart.yAxis.{x}.axisTick.interval .....	2825
6.2.70	chart.yAxis.{x}.axisTick.inside .....	2826
6.2.71	chart.yAxis.{x}.axisTick.length .....	2826
6.2.72	chart.yAxis.{x}.axisTick.lineStyle.object .....	2826
6.2.73	chart.yAxis.{x}.axisTick.lineStyle.color .....	2827

6.2.74	chart.yAxis.{x}.axisTick.lineStyle.width .....	2827
6.2.75	chart.yAxis.{x}.axisTick.lineStyle.type .....	2827
	Erlaubte Werte .....	2828
6.2.76	chart.yAxis.{x}.axisTick.lineStyle.shadowBlur .....	2828
6.2.77	chart.yAxis.{x}.axisTick.lineStyle.shadowColor .....	2828
6.2.78	chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetX .....	2829
6.2.79	chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetY .....	2829
6.2.80	chart.yAxis.{x}.axisTick.lineStyle.opacity .....	2829
6.2.81	chart.yAxis.{x}.axisLabel.object .....	2830
6.2.82	chart.yAxis.{x}.axisLabel.show .....	2830
6.2.83	chart.yAxis.{x}.axisLabel.interval .....	2830
6.2.84	chart.yAxis.{x}.axisLabel.inside .....	2831
6.2.85	chart.yAxis.{x}.axisLabel.rotate .....	2831
6.2.86	chart.yAxis.{x}.axisLabel.margin .....	2831
6.2.87	chart.yAxis.{x}.axisLabel.formatter .....	2832
6.2.88	chart.yAxis.{x}.axisLabel.showMinLabel .....	2832
6.2.89	chart.yAxis.{x}.axisLabel.showMaxLabel .....	2833
6.2.90	chart.yAxis.{x}.axisLabel.color .....	2833
6.2.91	chart.yAxis.{x}.axisLabel.fontStyle .....	2834
	Erlaubte Werte .....	2834
6.2.92	chart.yAxis.{x}.axisLabel.fontWeight .....	2834
	Erlaubte Werte .....	2835
6.2.93	chart.yAxis.{x}.axisLabel.fontFamily .....	2835
6.2.94	chart.yAxis.{x}.axisLabel.fontSize .....	2835
6.2.95	chart.yAxis.{x}.axisLabel.align .....	2836
	Erlaubte Werte .....	2836
6.2.96	chart.yAxis.{x}.axisLabel.verticalAlign .....	2836
	Erlaubte Werte .....	2837
6.2.97	chart.yAxis.{x}.axisLabel.lineHeight .....	2837
6.2.98	chart.yAxis.{x}.axisLabel.backgroundColor .....	2838
6.2.99	chart.yAxis.{x}.axisLabel.borderColor .....	2838
6.2.100	chart.yAxis.{x}.axisLabel.borderWidth .....	2839
6.2.101	chart.yAxis.{x}.axisLabel.borderRadius .....	2839
6.2.102	chart.yAxis.{x}.axisLabel.padding .....	2839
6.2.103	chart.yAxis.{x}.axisLabel.shadowColor .....	2840
6.2.104	chart.yAxis.{x}.axisLabel.shadowBlur .....	2840



6.2.105	chart.yAxis.{x}.axisLabel.shadowOffsetX .....	2841
6.2.106	chart.yAxis.{x}.axisLabel.shadowOffsetY .....	2841
6.2.107	chart.yAxis.{x}.axisLabel.width .....	2841
6.2.108	chart.yAxis.{x}.axisLabel.height .....	2842
6.2.109	chart.yAxis.{x}.axisLabel.textBorderColor .....	2842
6.2.110	chart.yAxis.{x}.axisLabel.textBorderWidth .....	2843
6.2.111	chart.yAxis.{x}.axisLabel.textShadowColor .....	2843
6.2.112	chart.yAxis.{x}.axisLabel.textShadowBlur .....	2843
6.2.113	chart.yAxis.{x}.axisLabel.textShadowOffsetX .....	2844
6.2.114	chart.yAxis.{x}.axisLabel.textShadowOffsetY .....	2844
6.2.115	chart.yAxis.{x}.axisLabel.rich.object .....	2844
6.2.116	chart.yAxis.{x}.splitLine.object .....	2845
6.2.117	chart.yAxis.{x}.splitLine.show .....	2846
6.2.118	chart.yAxis.{x}.splitLine.interval .....	2846
6.2.119	chart.yAxis.{x}.splitLine.lineStyle.object .....	2846
6.2.120	chart.yAxis.{x}.splitLine.lineStyle.color .....	2847
6.2.121	chart.yAxis.{x}.splitLine.lineStyle.width .....	2847
6.2.122	chart.yAxis.{x}.splitLine.lineStyle.type .....	2848
	Erlaubte Werte .....	2848
6.2.123	chart.yAxis.{x}.splitLine.lineStyle.shadowBlur .....	2848
6.2.124	chart.yAxis.{x}.splitLine.lineStyle.shadowColor .....	2849
6.2.125	chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetX .....	2849
6.2.126	chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetY .....	2849
6.2.127	chart.yAxis.{x}.splitLine.lineStyle.opacity .....	2850
6.2.128	chart.yAxis.{x}.splitArea.object .....	2850
6.2.129	chart.yAxis.{x}.splitArea.interval .....	2850
6.2.130	chart.yAxis.{x}.splitArea.show .....	2851
6.2.131	chart.yAxis.{x}.splitArea.areaStyle.object .....	2851
6.2.132	chart.yAxis.{x}.splitArea.areaStyle.color .....	2851
6.2.133	chart.yAxis.{x}.splitArea.areaStyle.shadowBlur .....	2852
6.2.134	chart.yAxis.{x}.splitArea.areaStyle.shadowColor .....	2852
6.2.135	chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetX .....	2853
6.2.136	chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetY .....	2853
6.2.137	chart.yAxis.{x}.splitArea.areaStyle.opacity .....	2853
6.2.138	chart.yAxis.{x}.axisPointer.object .....	2854
6.2.139	chart.yAxis.{x}.axisPointer.show .....	2854

6.2.140	chart.yAxis.{x}.axisPointer.type.....	2855
	Erlaubte Werte .....	2855
6.2.141	chart.yAxis.{x}.axisPointer.snap .....	2855
6.2.142	chart.yAxis.{x}.axisPointer.z .....	2855
6.2.143	chart.yAxis.{x}.axisPointer.label.object .....	2856
6.2.144	chart.yAxis.{x}.axisPointer.label.show .....	2856
6.2.145	chart.yAxis.{x}.axisPointer.label.precision .....	2857
6.2.146	chart.yAxis.{x}.axisPointer.label.formatter.....	2857
6.2.147	chart.yAxis.{x}.axisPointer.label.margin .....	2858
6.2.148	chart.yAxis.{x}.axisPointer.label.color .....	2859
6.2.149	chart.yAxis.{x}.axisPointer.label.fontStyle.....	2859
	Erlaubte Werte .....	2859
6.2.150	chart.yAxis.{x}.axisPointer.label.fontWeight .....	2860
	Erlaubte Werte .....	2860
6.2.151	chart.yAxis.{x}.axisPointer.label.fontFamily.....	2860
6.2.152	chart.yAxis.{x}.axisPointer.label.fontSize .....	2861
6.2.153	chart.yAxis.{x}.axisPointer.label.lineHeight .....	2861
6.2.154	chart.yAxis.{x}.axisPointer.label.width .....	2862
6.2.155	chart.yAxis.{x}.axisPointer.label.height .....	2862
6.2.156	chart.yAxis.{x}.axisPointer.label.textBorderColor.....	2862
6.2.157	chart.yAxis.{x}.axisPointer.label.textBorderWidth .....	2863
6.2.158	chart.yAxis.{x}.axisPointer.label.textShadowColor.....	2863
6.2.159	chart.yAxis.{x}.axisPointer.label.textShadowBlur.....	2864
6.2.160	chart.yAxis.{x}.axisPointer.label.textShadowOffsetX.....	2864
6.2.161	chart.yAxis.{x}.axisPointer.label.textShadowOffsetY.....	2864
6.2.162	chart.yAxis.{x}.axisPointer.label.padding.....	2865
6.2.163	chart.yAxis.{x}.axisPointer.label.backgroundColor .....	2865
6.2.164	chart.yAxis.{x}.axisPointer.label.borderColor .....	2866
6.2.165	chart.yAxis.{x}.axisPointer.label.borderWidth .....	2866
6.2.166	chart.yAxis.{x}.axisPointer.label.shadowBlur.....	2866
6.2.167	chart.yAxis.{x}.axisPointer.label.shadowColor.....	2867
6.2.168	chart.yAxis.{x}.axisPointer.label.shadowOffsetX.....	2867
6.2.169	chart.yAxis.{x}.axisPointer.label.shadowOffsetY.....	2868
6.2.170	chart.yAxis.{x}.axisPointer.lineStyle.object .....	2868
6.2.171	chart.yAxis.{x}.axisPointer.lineStyle.color .....	2868
6.2.172	chart.yAxis.{x}.axisPointer.lineStyle.width .....	2870

6.2.173	chart.yAxis.{x}.axisPointer.lineStyle.type .....	2870
	Erlaubte Werte .....	2870
6.2.174	chart.yAxis.{x}.axisPointer.lineStyle.shadowBlur .....	2870
6.2.175	chart.yAxis.{x}.axisPointer.lineStyle.shadowColor .....	2871
6.2.176	chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetX .....	2871
6.2.177	chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetY .....	2872
6.2.178	chart.yAxis.{x}.axisPointer.lineStyle.opacity .....	2872
6.2.179	chart.yAxis.{x}.axisPointer.shadowStyle.object .....	2872
6.2.180	chart.yAxis.{x}.axisPointer.shadowStyle.color .....	2873
6.2.181	chart.yAxis.{x}.axisPointer.shadowStyle.shadowBlur .....	2874
6.2.182	chart.yAxis.{x}.axisPointer.shadowStyle.shadowColor .....	2875
6.2.183	chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetX .....	2875
6.2.184	chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetY .....	2876
6.2.185	chart.yAxis.{x}.axisPointer.shadowStyle.opacity .....	2876
6.2.186	chart.yAxis.{x}.axisPointer.triggerTooltip .....	2876
6.2.187	chart.yAxis.{x}.axisPointer.value .....	2877
6.2.188	chart.yAxis.{x}.axisPointer.status .....	2877
	Erlaubte Werte .....	2877
6.2.189	chart.yAxis.{x}.axisPointer.handle.object .....	2877
6.2.190	chart.yAxis.{x}.axisPointer.handle.show .....	2878
6.2.191	chart.yAxis.{x}.axisPointer.handle.icon .....	2878
6.2.192	chart.yAxis.{x}.axisPointer.handle.size .....	2879
6.2.193	chart.yAxis.{x}.axisPointer.handle.margin .....	2879
6.2.194	chart.yAxis.{x}.axisPointer.handle.color .....	2879
6.2.195	chart.yAxis.{x}.axisPointer.handle.throttle .....	2880
6.2.196	chart.yAxis.{x}.axisPointer.handle.shadowBlur .....	2880
6.2.197	chart.yAxis.{x}.axisPointer.handle.shadowColor .....	2881
6.2.198	chart.yAxis.{x}.axisPointer.handle.shadowOffsetX .....	2881
6.2.199	chart.yAxis.{x}.axisPointer.handle.shadowOffsetY .....	2881
6.2.200	chart.yAxis.{x}.zlevel .....	2882
6.2.201	chart.yAxis.{x}.z .....	2882
6.2.202	chart.xAxis.{x}.object .....	2883
6.2.203	chart.xAxis.{x}.id .....	2883
6.2.204	chart.xAxis.{x}.show .....	2883
6.2.205	chart.xAxis.{x}.gridIndex .....	2884
6.2.206	chart.xAxis.{x}.position .....	2884

	Erlaubte Werte .....	2884
6.2.207	chart.xAxis.{x}.offset .....	2885
6.2.208	chart.xAxis.{x}.type.....	2885
	Erlaubte Werte .....	2885
6.2.209	chart.xAxis.{x}.name.....	2886
6.2.210	chart.xAxis.{x}.nameLocation .....	2886
	Erlaubte Werte .....	2886
6.2.211	chart.xAxis.{x}.nameTextStyle.object .....	2886
6.2.212	chart.xAxis.{x}.nameTextStyle.color .....	2887
6.2.213	chart.xAxis.{x}.nameTextStyle.fontStyle.....	2887
	Erlaubte Werte .....	2887
6.2.214	chart.xAxis.{x}.nameTextStyle.fontWeight .....	2888
	Erlaubte Werte .....	2888
6.2.215	chart.xAxis.{x}.nameTextStyle.fontFamily.....	2888
6.2.216	chart.xAxis.{x}.nameTextStyle.fontSize .....	2889
6.2.217	chart.xAxis.{x}.nameTextStyle.align.....	2889
	Erlaubte Werte .....	2889
6.2.218	chart.xAxis.{x}.nameTextStyle.verticalAlign .....	2890
	Erlaubte Werte .....	2890
6.2.219	chart.xAxis.{x}.nameTextStyle.lineHeight .....	2890
6.2.220	chart.xAxis.{x}.nameTextStyle.backgroundColor .....	2891
6.2.221	chart.xAxis.{x}.nameTextStyle.borderColor .....	2891
6.2.222	chart.xAxis.{x}.nameTextStyle.borderWidth .....	2892
6.2.223	chart.xAxis.{x}.nameTextStyle.borderRadius .....	2892
6.2.224	chart.xAxis.{x}.nameTextStyle.padding .....	2893
6.2.225	chart.xAxis.{x}.nameTextStyle.shadowColor.....	2893
6.2.226	chart.xAxis.{x}.nameTextStyle.shadowBlur.....	2893
6.2.227	chart.xAxis.{x}.nameTextStyle.shadowOffsetX.....	2894
6.2.228	chart.xAxis.{x}.nameTextStyle.shadowOffsetY.....	2894
6.2.229	chart.xAxis.{x}.nameTextStyle.width .....	2894
6.2.230	chart.xAxis.{x}.nameTextStyle.height .....	2895
6.2.231	chart.xAxis.{x}.nameTextStyle.textBorderColor.....	2895
6.2.232	chart.xAxis.{x}.nameTextStyle.textBorderWidth.....	2896
6.2.233	chart.xAxis.{x}.nameTextStyle.textShadowColor.....	2896
6.2.234	chart.xAxis.{x}.nameTextStyle.textShadowBlur.....	2896
6.2.235	chart.xAxis.{x}.nameTextStyle.textShadowOffsetX.....	2897

6.2.236	chart.xAxis.{x}.nameTextStyle.textShadowOffsetY .....	2897
6.2.237	chart.xAxis.{x}.nameTextStyle.rich.object .....	2898
6.2.238	chart.xAxis.{x}.nameGap .....	2899
6.2.239	chart.xAxis.{x}.nameRotate .....	2899
6.2.240	chart.xAxis.{x}.inverse .....	2899
6.2.241	chart.xAxis.{x}.boundaryGap .....	2900
6.2.242	chart.xAxis.{x}.min .....	2900
6.2.243	chart.xAxis.{x}.max .....	2901
6.2.244	chart.xAxis.{x}.scale .....	2901
6.2.245	chart.xAxis.{x}.splitNumber .....	2902
6.2.246	chart.xAxis.{x}.minInterval .....	2902
6.2.247	chart.xAxis.{x}.interval .....	2902
6.2.248	chart.xAxis.{x}.logBase .....	2903
6.2.249	chart.xAxis.{x}.silent .....	2903
6.2.250	chart.xAxis.{x}.triggerEvent .....	2904
6.2.251	chart.xAxis.{x}.axisLine.object .....	2904
6.2.252	chart.xAxis.{x}.axisLine.show .....	2905
6.2.253	chart.xAxis.{x}.axisLine.onZero .....	2905
6.2.254	chart.xAxis.{x}.axisLine.onZeroAxisIndex .....	2905
6.2.255	chart.xAxis.{x}.axisLine.symbol .....	2906
	Erlaubte Werte .....	2906
6.2.256	chart.xAxis.{x}.axisLine.symbolSize .....	2906
6.2.257	chart.xAxis.{x}.axisLine.symbolOffset .....	2907
6.2.258	chart.xAxis.{x}.axisLine.lineStyle.object .....	2907
6.2.259	chart.xAxis.{x}.axisLine.lineStyle.color .....	2907
6.2.260	chart.xAxis.{x}.axisLine.lineStyle.width .....	2909
6.2.261	chart.xAxis.{x}.axisLine.lineStyle.type .....	2909
	Erlaubte Werte .....	2909
6.2.262	chart.xAxis.{x}.axisLine.lineStyle.shadowBlur .....	2909
6.2.263	chart.xAxis.{x}.axisLine.lineStyle.shadowColor .....	2910
6.2.264	chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetX .....	2910
6.2.265	chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetY .....	2911
6.2.266	chart.xAxis.{x}.axisLine.lineStyle.opacity .....	2911
6.2.267	chart.xAxis.{x}.axisTick.object .....	2911
6.2.268	chart.xAxis.{x}.axisTick.show .....	2912
6.2.269	chart.xAxis.{x}.axisTick.alignWithLabel .....	2912

6.2.270	chart.xAxis.{x}.axisTick.interval.....	2912
6.2.271	chart.xAxis.{x}.axisTick.inside.....	2913
6.2.272	chart.xAxis.{x}.axisTick.length.....	2913
6.2.273	chart.xAxis.{x}.axisTick.lineStyle.object .....	2913
6.2.274	chart.xAxis.{x}.axisTick.lineStyle.color .....	2914
6.2.275	chart.xAxis.{x}.axisTick.lineStyle.width .....	2914
6.2.276	chart.xAxis.{x}.axisTick.lineStyle.type.....	2914
	Erlaubte Werte .....	2915
6.2.277	chart.xAxis.{x}.axisTick.lineStyle.shadowBlur .....	2915
6.2.278	chart.xAxis.{x}.axisTick.lineStyle.shadowColor .....	2915
6.2.279	chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetX.....	2916
6.2.280	chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetY .....	2916
6.2.281	chart.xAxis.{x}.axisTick.lineStyle.opacity.....	2916
6.2.282	chart.xAxis.{x}.axisLabel.object.....	2917
6.2.283	chart.xAxis.{x}.axisLabel.show .....	2917
6.2.284	chart.xAxis.{x}.axisLabel.interval .....	2918
6.2.285	chart.xAxis.{x}.axisLabel.inside .....	2918
6.2.286	chart.xAxis.{x}.axisLabel.rotate .....	2918
6.2.287	chart.xAxis.{x}.axisLabel.margin .....	2919
6.2.288	chart.xAxis.{x}.axisLabel.formatter .....	2919
6.2.289	chart.xAxis.{x}.axisLabel.showMinLabel .....	2920
6.2.290	chart.xAxis.{x}.axisLabel.showMaxLabel .....	2920
6.2.291	chart.xAxis.{x}.axisLabel.color.....	2921
6.2.292	chart.xAxis.{x}.axisLabel.fontStyle .....	2921
	Erlaubte Werte .....	2922
6.2.293	chart.xAxis.{x}.axisLabel.fontWeight.....	2922
	Erlaubte Werte .....	2922
6.2.294	chart.xAxis.{x}.axisLabel.fontFamily .....	2922
6.2.295	chart.xAxis.{x}.axisLabel.fontSize.....	2923
6.2.296	chart.xAxis.{x}.axisLabel.align .....	2923
	Erlaubte Werte .....	2924
6.2.297	chart.xAxis.{x}.axisLabel.verticalAlign .....	2924
	Erlaubte Werte .....	2924
6.2.298	chart.xAxis.{x}.axisLabel.lineHeight .....	2925
6.2.299	chart.xAxis.{x}.axisLabel.backgroundColor .....	2925
6.2.300	chart.xAxis.{x}.axisLabel.borderColor.....	2926

6.2.301	chart.xAxis.{x}.axisLabel.borderWidth .....	2926
6.2.302	chart.xAxis.{x}.axisLabel.borderRadius.....	2927
6.2.303	chart.xAxis.{x}.axisLabel.padding .....	2927
6.2.304	chart.xAxis.{x}.axisLabel.shadowColor .....	2927
6.2.305	chart.xAxis.{x}.axisLabel.shadowBlur .....	2928
6.2.306	chart.xAxis.{x}.axisLabel.shadowOffsetX .....	2928
6.2.307	chart.xAxis.{x}.axisLabel.shadowOffsetY .....	2928
6.2.308	chart.xAxis.{x}.axisLabel.width.....	2929
6.2.309	chart.xAxis.{x}.axisLabel.height.....	2929
6.2.310	chart.xAxis.{x}.axisLabel.textBorderColor .....	2930
6.2.311	chart.xAxis.{x}.axisLabel.textBorderWidth .....	2930
6.2.312	chart.xAxis.{x}.axisLabel.textShadowColor .....	2930
6.2.313	chart.xAxis.{x}.axisLabel.textShadowBlur .....	2931
6.2.314	chart.xAxis.{x}.axisLabel.textShadowOffsetX .....	2931
6.2.315	chart.xAxis.{x}.axisLabel.textShadowOffsetY .....	2932
6.2.316	chart.xAxis.{x}.axisLabel.rich.object .....	2932
6.2.317	chart.xAxis.{x}.splitLine.object .....	2933
6.2.318	chart.xAxis.{x}.splitLine.show.....	2934
6.2.319	chart.xAxis.{x}.splitLine.interval.....	2934
6.2.320	chart.xAxis.{x}.splitLine.lineStyle.object .....	2934
6.2.321	chart.xAxis.{x}.splitLine.lineStyle.color .....	2935
6.2.322	chart.xAxis.{x}.splitLine.lineStyle.width .....	2935
6.2.323	chart.xAxis.{x}.splitLine.lineStyle.type.....	2936
	Erlaubte Werte .....	2936
6.2.324	chart.xAxis.{x}.splitLine.lineStyle.shadowBlur .....	2936
6.2.325	chart.xAxis.{x}.splitLine.lineStyle.shadowColor.....	2937
6.2.326	chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetX.....	2937
6.2.327	chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetY .....	2937
6.2.328	chart.xAxis.{x}.splitLine.lineStyle.opacity.....	2938
6.2.329	chart.xAxis.{x}.splitArea.object.....	2938
6.2.330	chart.xAxis.{x}.splitArea.interval .....	2938
6.2.331	chart.xAxis.{x}.splitArea.show .....	2939
6.2.332	chart.xAxis.{x}.splitArea.areaStyle.object.....	2939
6.2.333	chart.xAxis.{x}.splitArea.areaStyle.color.....	2939
6.2.334	chart.xAxis.{x}.splitArea.areaStyle.shadowBlur .....	2940
6.2.335	chart.xAxis.{x}.splitArea.areaStyle.shadowColor .....	2940

6.2.336	chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetX .....	2941
6.2.337	chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetY .....	2941
6.2.338	chart.xAxis.{x}.splitArea.areaStyle.opacity .....	2941
6.2.339	chart.xAxis.{x}.axisPointer.object .....	2942
6.2.340	chart.xAxis.{x}.axisPointer.show .....	2942
6.2.341	chart.xAxis.{x}.axisPointer.type .....	2942
	Erlaubte Werte .....	2943
6.2.342	chart.xAxis.{x}.axisPointer.snap .....	2943
6.2.343	chart.xAxis.{x}.axisPointer.z .....	2943
6.2.344	chart.xAxis.{x}.axisPointer.label.object .....	2944
6.2.345	chart.xAxis.{x}.axisPointer.label.show .....	2944
6.2.346	chart.xAxis.{x}.axisPointer.label.precision .....	2944
6.2.347	chart.xAxis.{x}.axisPointer.label.formatter .....	2945
6.2.348	chart.xAxis.{x}.axisPointer.label.margin .....	2946
6.2.349	chart.xAxis.{x}.axisPointer.label.color .....	2947
6.2.350	chart.xAxis.{x}.axisPointer.label.fontStyle .....	2947
	Erlaubte Werte .....	2947
6.2.351	chart.xAxis.{x}.axisPointer.label.fontWeight .....	2948
	Erlaubte Werte .....	2948
6.2.352	chart.xAxis.{x}.axisPointer.label.fontFamily .....	2948
6.2.353	chart.xAxis.{x}.axisPointer.label.fontSize .....	2949
6.2.354	chart.xAxis.{x}.axisPointer.label.lineHeight .....	2949
6.2.355	chart.xAxis.{x}.axisPointer.label.width .....	2950
6.2.356	chart.xAxis.{x}.axisPointer.label.height .....	2950
6.2.357	chart.xAxis.{x}.axisPointer.label.textBorderColor .....	2950
6.2.358	chart.xAxis.{x}.axisPointer.label.textBorderWidth .....	2951
6.2.359	chart.xAxis.{x}.axisPointer.label.textShadowColor .....	2951
6.2.360	chart.xAxis.{x}.axisPointer.label.textShadowBlur .....	2952
6.2.361	chart.xAxis.{x}.axisPointer.label.textShadowOffsetX .....	2952
6.2.362	chart.xAxis.{x}.axisPointer.label.textShadowOffsetY .....	2952
6.2.363	chart.xAxis.{x}.axisPointer.label.padding .....	2953
6.2.364	chart.xAxis.{x}.axisPointer.label.backgroundColor .....	2953
6.2.365	chart.xAxis.{x}.axisPointer.label.borderColor .....	2954
6.2.366	chart.xAxis.{x}.axisPointer.label.borderWidth .....	2954
6.2.367	chart.xAxis.{x}.axisPointer.label.shadowBlur .....	2954
6.2.368	chart.xAxis.{x}.axisPointer.label.shadowColor .....	2955



6.2.369	chart.xAxis.{x}.axisPointer.label.shadowOffsetX.....	2955
6.2.370	chart.xAxis.{x}.axisPointer.label.shadowOffsetY.....	2956
6.2.371	chart.xAxis.{x}.axisPointer.lineStyle.object.....	2956
6.2.372	chart.xAxis.{x}.axisPointer.lineStyle.color.....	2956
6.2.373	chart.xAxis.{x}.axisPointer.lineStyle.width.....	2958
6.2.374	chart.xAxis.{x}.axisPointer.lineStyle.type.....	2958
	Erlaubte Werte.....	2958
6.2.375	chart.xAxis.{x}.axisPointer.lineStyle.shadowBlur.....	2958
6.2.376	chart.xAxis.{x}.axisPointer.lineStyle.shadowColor.....	2959
6.2.377	chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetX.....	2959
6.2.378	chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetY.....	2960
6.2.379	chart.xAxis.{x}.axisPointer.lineStyle.opacity.....	2960
6.2.380	chart.xAxis.{x}.axisPointer.shadowStyle.object.....	2960
6.2.381	chart.xAxis.{x}.axisPointer.shadowStyle.color.....	2961
6.2.382	chart.xAxis.{x}.axisPointer.shadowStyle.shadowBlur.....	2962
6.2.383	chart.xAxis.{x}.axisPointer.shadowStyle.shadowColor.....	2963
6.2.384	chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetX.....	2963
6.2.385	chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetY.....	2964
6.2.386	chart.xAxis.{x}.axisPointer.shadowStyle.opacity.....	2964
6.2.387	chart.xAxis.{x}.axisPointer.triggerTooltip.....	2964
6.2.388	chart.xAxis.{x}.axisPointer.value.....	2965
6.2.389	chart.xAxis.{x}.axisPointer.status.....	2965
	Erlaubte Werte.....	2965
6.2.390	chart.xAxis.{x}.axisPointer.handle.object.....	2965
6.2.391	chart.xAxis.{x}.axisPointer.handle.show.....	2966
6.2.392	chart.xAxis.{x}.axisPointer.handle.icon.....	2966
6.2.393	chart.xAxis.{x}.axisPointer.handle.size.....	2967
6.2.394	chart.xAxis.{x}.axisPointer.handle.margin.....	2967
6.2.395	chart.xAxis.{x}.axisPointer.handle.color.....	2967
6.2.396	chart.xAxis.{x}.axisPointer.handle.throttle.....	2968
6.2.397	chart.xAxis.{x}.axisPointer.handle.shadowBlur.....	2968
6.2.398	chart.xAxis.{x}.axisPointer.handle.shadowColor.....	2969
6.2.399	chart.xAxis.{x}.axisPointer.handle.shadowOffsetX.....	2969
6.2.400	chart.xAxis.{x}.axisPointer.handle.shadowOffsetY.....	2969
6.2.401	chart.xAxis.{x}.zlevel.....	2970
6.2.402	chart.xAxis.{x}.z.....	2970

6.2.403	chart.axis.flip.....	2971
<b>6.3</b>	<b>ECharts (5) Axis Pointer.....</b>	<b>2971</b>
6.3.1	chart.axisPointer.object .....	2971
6.3.2	chart.axisPointer.id.....	2972
6.3.3	chart.axisPointer.show .....	2973
6.3.4	chart.axisPointer.type .....	2973
	Erlaubte Werte .....	2973
6.3.5	chart.axisPointer.snap.....	2974
6.3.6	chart.axisPointer.z .....	2974
6.3.7	chart.axisPointer.label.object .....	2974
6.3.8	chart.axisPointer.label.show.....	2975
6.3.9	chart.axisPointer.label.precision .....	2975
6.3.10	chart.axisPointer.label.formatter .....	2975
6.3.11	chart.axisPointer.label.margin.....	2977
6.3.12	chart.axisPointer.label.color .....	2977
6.3.13	chart.axisPointer.label.fontStyle .....	2977
	Erlaubte Werte .....	2978
6.3.14	chart.axisPointer.label.fontWeight .....	2978
	Erlaubte Werte .....	2979
6.3.15	chart.axisPointer.label.fontFamily .....	2979
6.3.16	chart.axisPointer.label.fontSize .....	2979
6.3.17	chart.axisPointer.label.lineHeight .....	2979
6.3.18	chart.axisPointer.label.width .....	2980
6.3.19	chart.axisPointer.label.height.....	2981
6.3.20	chart.axisPointer.label.textBorderColor .....	2981
6.3.21	chart.axisPointer.label.textBorderWidth.....	2981
6.3.22	chart.axisPointer.label.textShadowColor .....	2982
6.3.23	chart.axisPointer.label.textShadowBlur .....	2982
6.3.24	chart.axisPointer.label.textShadowOffsetX .....	2982
6.3.25	chart.axisPointer.label.textShadowOffsetY .....	2983
6.3.26	chart.axisPointer.label.padding.....	2983
6.3.27	chart.axisPointer.label.backgroundColor .....	2984
6.3.28	chart.axisPointer.label.borderColor .....	2984
6.3.29	chart.axisPointer.label.borderWidth .....	2985
6.3.30	chart.axisPointer.label.shadowBlur .....	2985
6.3.31	chart.axisPointer.label.shadowColor .....	2985

6.3.32	chart.axisPointer.label.shadowOffsetX .....	2986
6.3.33	chart.axisPointer.label.shadowOffsetY .....	2986
6.3.34	chart.axisPointer.lineStyle.object.....	2987
6.3.35	chart.axisPointer.lineStyle.color.....	2987
6.3.36	chart.axisPointer.lineStyle.width.....	2988
6.3.37	chart.axisPointer.lineStyle.type .....	2989
	Erlaubte Werte .....	2989
6.3.38	chart.axisPointer.lineStyle.shadowBlur .....	2989
6.3.39	chart.axisPointer.lineStyle.shadowColor .....	2990
6.3.40	chart.axisPointer.lineStyle.shadowOffsetX.....	2990
6.3.41	chart.axisPointer.lineStyle.shadowOffsetY .....	2990
6.3.42	chart.axisPointer.lineStyle.opacity.....	2991
6.3.43	chart.axisPointer.shadowStyle.object.....	2991
6.3.44	chart.axisPointer.shadowStyle.color.....	2991
6.3.45	chart.axisPointer.shadowStyle.shadowBlur .....	2993
6.3.46	chart.axisPointer.shadowStyle.shadowColor .....	2994
6.3.47	chart.axisPointer.shadowStyle.shadowOffsetX .....	2994
6.3.48	chart.axisPointer.shadowStyle.shadowOffsetY .....	2995
6.3.49	chart.axisPointer.shadowStyle.opacity.....	2995
6.3.50	chart.axisPointer.triggerTooltip.....	2995
6.3.51	chart.axisPointer.value.....	2996
6.3.52	chart.axisPointer.status.....	2996
	Erlaubte Werte .....	2996
6.3.53	chart.axisPointer.handle.object.....	2996
6.3.54	chart.axisPointer.handle.show .....	2997
6.3.55	chart.axisPointer.handle.icon .....	2997
6.3.56	chart.axisPointer.handle.size.....	2998
6.3.57	chart.axisPointer.handle.margin .....	2998
6.3.58	chart.axisPointer.handle.color.....	2998
6.3.59	chart.axisPointer.handle.throttle .....	2999
6.3.60	chart.axisPointer.handle.shadowBlur .....	2999
6.3.61	chart.axisPointer.handle.shadowColor .....	3000
6.3.62	chart.axisPointer.handle.shadowOffsetX.....	3000
6.3.63	chart.axisPointer.handle.shadowOffsetY .....	3000
6.3.64	chart.axisPointer.link.....	3001
6.3.65	chart.axisPointer.triggerOn.....	3002

	Erlaubte Werte .....	3003
6.3.66	chart.axisPointer.label.textBorderStyle .....	3003
	Erlaubte Werte .....	3004
6.3.67	chart.axisPointer.label.textBorderDashOffset .....	3004
6.3.68	chart.axisPointer.label.overflow .....	3004
	Erlaubte Werte .....	3004
6.3.69	chart.axisPointer.lineStyle.dashOffset .....	3005
6.3.70	chart.axisPointer.lineStyle.cap .....	3005
	Erlaubte Werte .....	3005
6.3.71	chart.axisPointer.lineStyle.join .....	3005
	Erlaubte Werte .....	3006
6.3.72	chart.axisPointer.lineStyle.miterLimit.....	3006
6.4	ECharts (5) Brush.....	3007
6.4.1	chart.brush.object .....	3007
6.4.2	chart.brush.id.....	3007
6.4.3	chart.brush.toolbox .....	3007
	Erlaubte Werte .....	3008
6.4.4	chart.brush.brushLink .....	3008
6.4.5	chart.brush.seriesIndex .....	3009
6.4.6	chart.brush.geoIndex.....	3009
6.4.7	chart.brush.xAxisIndex .....	3010
6.4.8	chart.brush.yAxisIndex .....	3012
6.4.9	chart.brush.brushType .....	3013
	Erlaubte Werte .....	3013
6.4.10	chart.brush.brushMode .....	3014
	Erlaubte Werte .....	3014
6.4.11	chart.brush.transformable .....	3014
6.4.12	chart.brush.brushStyle .....	3014
6.4.13	chart.brush.throttleType .....	3015
6.4.14	chart.brush.throttleDelay.....	3015
6.4.15	chart.brush.removeOnClick .....	3016
6.4.16	chart.brush.inBrush .....	3016
6.4.17	chart.brush.outOfBrush.....	3017
6.4.18	chart.brush.z .....	3018
6.5	ECharts (5) Calendar .....	3018

6.5.1	chart.calendar.object .....	3018
6.5.2	chart.calendar.id .....	3019
6.5.3	chart.calendar.zlevel .....	3019
6.5.4	chart.calendar.z .....	3020
6.5.5	chart.calendar.left .....	3020
	Erlaubte Werte .....	3020
6.5.6	chart.calendar.top .....	3021
	Erlaubte Werte .....	3021
6.5.7	chart.calendar.right .....	3021
	Erlaubte Werte .....	3022
6.5.8	chart.calendar.bottom .....	3022
	Erlaubte Werte .....	3022
6.5.9	chart.calendar.width .....	3022
6.5.10	chart.calendar.height .....	3023
6.5.11	chart.calendar.range .....	3023
6.5.12	chart.calendar.cellSize .....	3024
6.5.13	chart.calendar.orient.....	3025
	Erlaubte Werte .....	3025
6.5.14	chart.calendar.splitLine.object .....	3025
6.5.15	chart.calendar.splitLine.show.....	3025
6.5.16	chart.calendar.splitLine.lineStyle.object .....	3026
6.5.17	chart.calendar.splitLine.lineStyle.color .....	3026
6.5.18	chart.calendar.splitLine.lineStyle.width .....	3027
6.5.19	chart.calendar.splitLine.lineStyle.type.....	3028
	Erlaubte Werte .....	3028
6.5.20	chart.calendar.splitLine.lineStyle.shadowBlur .....	3028
6.5.21	chart.calendar.splitLine.lineStyle.shadowColor.....	3029
6.5.22	chart.calendar.splitLine.lineStyle.shadowOffsetX.....	3029
6.5.23	chart.calendar.splitLine.lineStyle.shadowOffsetY .....	3029
6.5.24	chart.calendar.splitLine.lineStyle.opacity.....	3030
6.5.25	chart.calendar.itemStyle.object .....	3030
6.5.26	chart.calendar.itemStyle.color .....	3030
6.5.27	chart.calendar.itemStyle.borderColor .....	3032
6.5.28	chart.calendar.itemStyle.borderWidth.....	3033
6.5.29	chart.calendar.itemStyle.borderType .....	3033
	Erlaubte Werte .....	3033

6.5.30	chart.calendar.itemStyle.shadowBlur .....	3034
6.5.31	chart.calendar.itemStyle.shadowColor .....	3034
6.5.32	chart.calendar.itemStyle.shadowOffsetX .....	3034
6.5.33	chart.calendar.itemStyle.shadowOffsetY .....	3035
6.5.34	chart.calendar.itemStyle.opacity .....	3035
6.5.35	chart.calendar.dayLabel.object .....	3035
6.5.36	chart.calendar.dayLabel.show .....	3036
6.5.37	chart.calendar.dayLabel.firstDay .....	3036
6.5.38	chart.calendar.dayLabel.margin .....	3037
6.5.39	chart.calendar.dayLabel.position .....	3037
	Erlaubte Werte .....	3037
6.5.40	chart.calendar.dayLabel.nameMap .....	3038
6.5.41	chart.calendar.dayLabel.color .....	3038
6.5.42	chart.calendar.dayLabel.fontStyle .....	3039
	Erlaubte Werte .....	3039
6.5.43	chart.calendar.dayLabel.fontWeight .....	3039
	Erlaubte Werte .....	3040
6.5.44	chart.calendar.dayLabel.fontFamily .....	3040
6.5.45	chart.calendar.dayLabel.fontSize .....	3040
6.5.46	chart.calendar.dayLabel.align .....	3040
	Erlaubte Werte .....	3041
6.5.47	chart.calendar.dayLabel.verticalAlign .....	3041
	Erlaubte Werte .....	3042
6.5.48	chart.calendar.dayLabel.lineHeight .....	3042
6.5.49	chart.calendar.dayLabel.backgroundColor .....	3042
6.5.50	chart.calendar.dayLabel.borderColor .....	3043
6.5.51	chart.calendar.dayLabel.borderWidth .....	3043
6.5.52	chart.calendar.dayLabel.borderRadius .....	3044
6.5.53	chart.calendar.dayLabel.padding .....	3044
6.5.54	chart.calendar.dayLabel.shadowColor .....	3045
6.5.55	chart.calendar.dayLabel.shadowBlur .....	3045
6.5.56	chart.calendar.dayLabel.shadowOffsetX .....	3045
6.5.57	chart.calendar.dayLabel.shadowOffsetY .....	3046
6.5.58	chart.calendar.dayLabel.width .....	3046
6.5.59	chart.calendar.dayLabel.height .....	3047
6.5.60	chart.calendar.dayLabel.textBorderColor .....	3047

6.5.61	chart.calendar.dayLabel.textBorderWidth.....	3047
6.5.62	chart.calendar.dayLabel.textShadowColor .....	3048
6.5.63	chart.calendar.dayLabel.textShadowBlur.....	3048
6.5.64	chart.calendar.dayLabel.textShadowOffsetX .....	3048
6.5.65	chart.calendar.dayLabel.textShadowOffsetY.....	3049
6.5.66	chart.calendar.dayLabel.rich.object.....	3049
6.5.67	chart.calendar.monthLabel.object .....	3050
6.5.68	chart.calendar.monthLabel.show.....	3051
6.5.69	chart.calendar.monthLabel.align .....	3051
	Erlaubte Werte .....	3051
6.5.70	chart.calendar.monthLabel.margin.....	3052
6.5.71	chart.calendar.monthLabel.position .....	3052
	Erlaubte Werte .....	3052
6.5.72	chart.calendar.monthLabel.nameMap .....	3052
6.5.73	chart.calendar.monthLabel.formatter .....	3053
6.5.74	chart.calendar.monthLabel.color .....	3054
6.5.75	chart.calendar.monthLabel.fontStyle .....	3055
	Erlaubte Werte .....	3055
6.5.76	chart.calendar.monthLabel.fontWeight .....	3055
	Erlaubte Werte .....	3056
6.5.77	chart.calendar.monthLabel.fontFamily .....	3056
6.5.78	chart.calendar.monthLabel.fontSize.....	3056
6.5.79	chart.calendar.monthLabel.verticalAlign.....	3056
	Erlaubte Werte .....	3057
6.5.80	chart.calendar.monthLabel.lineHeight .....	3057
6.5.81	chart.calendar.monthLabel.backgroundColor .....	3058
6.5.82	chart.calendar.monthLabel.borderColor .....	3058
6.5.83	chart.calendar.monthLabel.borderWidth .....	3059
6.5.84	chart.calendar.monthLabel.borderRadius.....	3059
6.5.85	chart.calendar.monthLabel.padding.....	3059
6.5.86	chart.calendar.monthLabel.shadowColor .....	3060
6.5.87	chart.calendar.monthLabel.shadowBlur .....	3060
6.5.88	chart.calendar.monthLabel.shadowOffsetX .....	3061
6.5.89	chart.calendar.monthLabel.shadowOffsetY .....	3061
6.5.90	chart.calendar.monthLabel.width .....	3061
6.5.91	chart.calendar.monthLabel.height.....	3062

6.5.92	chart.calendar.monthLabel.textBorderColor .....	3062
6.5.93	chart.calendar.monthLabel.textBorderWidth.....	3063
6.5.94	chart.calendar.monthLabel.textShadowColor .....	3063
6.5.95	chart.calendar.monthLabel.textShadowBlur .....	3063
6.5.96	chart.calendar.monthLabel.textShadowOffsetX .....	3064
6.5.97	chart.calendar.monthLabel.textShadowOffsetY .....	3064
6.5.98	chart.calendar.monthLabel.rich.object.....	3064
6.5.99	chart.calendar.yearLabel.object .....	3065
6.5.100	chart.calendar.yearLabel.show.....	3066
6.5.101	chart.calendar.yearLabel.margin.....	3066
6.5.102	chart.calendar.yearLabel.position.....	3066
	Erlaubte Werte .....	3067
6.5.103	chart.calendar.yearLabel.formatter .....	3067
6.5.104	chart.calendar.yearLabel.color .....	3068
6.5.105	chart.calendar.yearLabel.fontStyle .....	3068
	Erlaubte Werte .....	3069
6.5.106	chart.calendar.yearLabel.fontWeight.....	3069
	Erlaubte Werte .....	3069
6.5.107	chart.calendar.yearLabel.fontFamily .....	3069
6.5.108	chart.calendar.yearLabel.fontSize.....	3070
6.5.109	chart.calendar.yearLabel.align .....	3070
	Erlaubte Werte .....	3071
6.5.110	chart.calendar.yearLabel.verticalAlign.....	3071
	Erlaubte Werte .....	3071
6.5.111	chart.calendar.yearLabel.lineHeight .....	3072
6.5.112	chart.calendar.yearLabel.backgroundColor .....	3072
6.5.113	chart.calendar.yearLabel.borderColor .....	3073
6.5.114	chart.calendar.yearLabel.borderWidth .....	3073
6.5.115	chart.calendar.yearLabel.borderRadius.....	3074
6.5.116	chart.calendar.yearLabel.padding.....	3074
6.5.117	chart.calendar.yearLabel.shadowColor .....	3074
6.5.118	chart.calendar.yearLabel.shadowBlur .....	3075
6.5.119	chart.calendar.yearLabel.shadowOffsetX .....	3075
6.5.120	chart.calendar.yearLabel.shadowOffsetY .....	3075
6.5.121	chart.calendar.yearLabel.width.....	3076
6.5.122	chart.calendar.yearLabel.height.....	3076



6.5.123	chart.calendar.yearLabel.textBorderColor .....	3077
6.5.124	chart.calendar.yearLabel.textBorderWidth.....	3077
6.5.125	chart.calendar.yearLabel.textShadowColor .....	3077
6.5.126	chart.calendar.yearLabel.textShadowBlur .....	3078
6.5.127	chart.calendar.yearLabel.textShadowOffsetX .....	3078
6.5.128	chart.calendar.yearLabel.textShadowOffsetY .....	3079
6.5.129	chart.calendar.yearLabel.rich.object.....	3079
6.5.130	chart.calendar.silent.....	3080
<b>6.6</b>	<b>ECharts (5) Data.....</b>	<b>3081</b>
6.6.1	chart.data.hierachy.render .....	3081
	Erlaubte Werte .....	3081
6.6.2	chart.dynamicSum.show.....	3081
6.6.3	chart.dynamicSum.position.....	3082
6.6.4	chart.dynamicSum.color .....	3082
6.6.5	chart.dynamicSum.roundedDigits.....	3082
6.6.6	chart.dynamicSum.excludeSeriesList .....	3083
6.6.7	chart.dynamicSum.retainStyleSeriesList .....	3083
6.6.8	chart.dynamicSum.retainStyleSeriesList .....	3084
6.6.9	chart.dynamicSum.customSeriesHandlerBlock .....	3084
6.6.10	chart.dynamicSum.customHandleLegendSelectChangedBlock .....	3085
<b>6.7</b>	<b>ECharts (5) Data Zoom .....</b>	<b>3085</b>
6.7.1	chart.dataZoom.object.....	3085
6.7.2	chart.dataZoom.type.....	3086
	Erlaubte Werte .....	3086
6.7.3	chart.dataZoom.id .....	3086
6.7.4	chart.dataZoom.show .....	3086
6.7.5	chart.dataZoom.backgroundColor .....	3087
6.7.6	chart.dataZoom.dataBackground.object.....	3087
6.7.7	chart.dataZoom.dataBackground.lineStyle.object .....	3087
6.7.8	chart.dataZoom.dataBackground.lineStyle.color .....	3088
6.7.9	chart.dataZoom.dataBackground.lineStyle.width .....	3089
6.7.10	chart.dataZoom.dataBackground.lineStyle.type .....	3090
	Erlaubte Werte .....	3090
6.7.11	chart.dataZoom.dataBackground.lineStyle.shadowBlur .....	3090
6.7.12	chart.dataZoom.dataBackground.lineStyle.shadowColor .....	3091

6.7.13	chart.dataZoom.dataBackground.lineStyle.shadowOffsetX .....	3091
6.7.14	chart.dataZoom.dataBackground.lineStyle.shadowOffsetY .....	3091
6.7.15	chart.dataZoom.dataBackground.lineStyle.opacity .....	3092
6.7.16	chart.dataZoom.dataBackground.areaStyle.object .....	3092
6.7.17	chart.dataZoom.dataBackground.areaStyle.color .....	3092
6.7.18	chart.dataZoom.dataBackground.areaStyle.shadowBlur .....	3094
6.7.19	chart.dataZoom.dataBackground.areaStyle.shadowColor .....	3095
6.7.20	chart.dataZoom.dataBackground.areaStyle.shadowOffsetX .....	3095
6.7.21	chart.dataZoom.dataBackground.areaStyle.shadowOffsetY .....	3096
6.7.22	chart.dataZoom.dataBackground.areaStyle.opacity .....	3096
6.7.23	chart.dataZoom.fillerColor .....	3096
6.7.24	chart.dataZoom.borderColor .....	3097
6.7.25	chart.dataZoom.handleIcon .....	3097
6.7.26	chart.dataZoom.handleSize .....	3098
	Erlaubte Werte .....	3098
6.7.27	chart.dataZoom.handleStyle.object .....	3098
6.7.28	chart.dataZoom.handleStyle.color .....	3098
6.7.29	chart.dataZoom.handleStyle.borderColor .....	3100
6.7.30	chart.dataZoom.handleStyle.borderWidth .....	3100
6.7.31	chart.dataZoom.handleStyle.borderType .....	3100
	Erlaubte Werte .....	3101
6.7.32	chart.dataZoom.handleStyle.shadowBlur .....	3101
6.7.33	chart.dataZoom.handleStyle.shadowColor .....	3101
6.7.34	chart.dataZoom.handleStyle.shadowOffsetX .....	3102
6.7.35	chart.dataZoom.handleStyle.shadowOffsetY .....	3102
6.7.36	chart.dataZoom.handleStyle.opacity .....	3102
6.7.37	chart.dataZoom.labelPrecision .....	3103
6.7.38	chart.dataZoom.labelFormatter .....	3103
6.7.39	chart.dataZoom.showDetail .....	3104
6.7.40	chart.dataZoom.showDataShadow .....	3104
6.7.41	chart.dataZoom.realtime .....	3105
6.7.42	chart.dataZoom.textStyle.object .....	3105
6.7.43	chart.dataZoom.textStyle.color .....	3105
6.7.44	chart.dataZoom.textStyle.fontStyle .....	3106
	Erlaubte Werte .....	3106
6.7.45	chart.dataZoom.textStyle.fontWeight .....	3106

	Erlaubte Werte .....	3107
6.7.46	chart.dataZoom.textStyle.fontFamily .....	3107
6.7.47	chart.dataZoom.textStyle.fontSize.....	3107
6.7.48	chart.dataZoom.textStyle.lineHeight .....	3107
6.7.49	chart.dataZoom.textStyle.width.....	3108
6.7.50	chart.dataZoom.textStyle.height.....	3109
6.7.51	chart.dataZoom.textStyle.textBorderColor .....	3109
6.7.52	chart.dataZoom.textStyle.textBorderWidth .....	3109
6.7.53	chart.dataZoom.textStyle.textShadowColor .....	3110
6.7.54	chart.dataZoom.textStyle.textShadowBlur .....	3110
6.7.55	chart.dataZoom.textStyle.textShadowOffsetX .....	3110
6.7.56	chart.dataZoom.textStyle.textShadowOffsetY .....	3111
6.7.57	chart.dataZoom.xAxisIndex.....	3111
6.7.58	chart.dataZoom.yAxisIndex .....	3112
6.7.59	chart.dataZoom.radiusAxisIndex.....	3113
6.7.60	chart.dataZoom.angleAxisIndex .....	3114
6.7.61	chart.dataZoom.filterMode .....	3115
	Erlaubte Werte .....	3117
6.7.62	chart.dataZoom.start .....	3117
6.7.63	chart.dataZoom.end.....	3118
6.7.64	chart.dataZoom.startValue .....	3118
6.7.65	chart.dataZoom.endValue.....	3118
6.7.66	chart.dataZoom.minSpan .....	3119
6.7.67	chart.dataZoom.maxSpan .....	3119
6.7.68	chart.dataZoom.minValueSpan.....	3120
6.7.69	chart.dataZoom.maxValueSpan .....	3120
6.7.70	chart.dataZoom.orient .....	3121
	Erlaubte Werte .....	3121
6.7.71	chart.dataZoom.zoomLock.....	3121
6.7.72	chart.dataZoom.throttle .....	3121
6.7.73	chart.dataZoom.rangeMode .....	3122
	Erlaubte Werte .....	3122
6.7.74	chart.dataZoom.zlevel.....	3122
6.7.75	chart.dataZoom.z.....	3123
6.7.76	chart.dataZoom.left.....	3123
	Erlaubte Werte .....	3124

6.7.77	chart.dataZoom.top .....	3124
	Erlaubte Werte .....	3124
6.7.78	chart.dataZoom.right .....	3124
	Erlaubte Werte .....	3125
6.7.79	chart.dataZoom.bottom.....	3125
	Erlaubte Werte .....	3125
6.7.80	chart.dataZoom.disabled.....	3126
6.7.81	chart.dataZoom.zoomOnMouseWheel.....	3126
	Erlaubte Werte .....	3126
6.7.82	chart.dataZoom.moveOnMouseMove .....	3127
	Erlaubte Werte .....	3127
6.7.83	chart.dataZoom.moveOnMouseWheel.....	3127
	Erlaubte Werte .....	3128
6.7.84	chart.dataZoom.preventDefaultMouseMove .....	3128
6.7.85	chart.dataZoom.dataBackground.lineStyle.dashOffset .....	3128
6.7.86	chart.dataZoom.dataBackground.lineStyle.cap.....	3128
	Erlaubte Werte .....	3129
6.7.87	chart.dataZoom.dataBackground.lineStyle.join .....	3129
	Erlaubte Werte .....	3130
6.7.88	chart.dataZoom.dataBackground.lineStyle.miterLimit.....	3130
6.7.89	chart.dataZoom.handleStyle.borderDashOffset .....	3130
6.7.90	chart.dataZoom.handleStyle.borderCap .....	3130
	Erlaubte Werte .....	3131
6.7.91	chart.dataZoom.handleStyle.borderJoin .....	3131
	Erlaubte Werte .....	3132
6.7.92	chart.dataZoom.handleStyle.borderMiterLimit.....	3132
6.7.93	chart.dataZoom.textStyle.textBorderStyle .....	3132
	Erlaubte Werte .....	3133
6.7.94	chart.dataZoom.textStyle.textBorderDashOffset .....	3133
6.7.95	chart.dataZoom.textStyle.overflow.....	3133
	Erlaubte Werte .....	3133
6.7.96	chart.dataZoom.textStyle.ellipsis.....	3134
6.7.97	chart.dataZoom.selectedDataBackground.object .....	3134
6.7.98	chart.dataZoom.selectedDataBackground.lineStyle.object.....	3134
6.7.99	chart.dataZoom.selectedDataBackground.lineStyle.color.....	3135
6.7.100	chart.dataZoom.selectedDataBackground.lineStyle.width.....	3136

6.7.101	chart.dataZoom.selectedDataBackground.lineStyle.type .....	3137
	Erlaubte Werte .....	3137
6.7.102	chart.dataZoom.selectedDataBackground.lineStyle.dashOffset .....	3137
6.7.103	chart.dataZoom.selectedDataBackground.lineStyle.cap .....	3137
	Erlaubte Werte .....	3138
6.7.104	chart.dataZoom.selectedDataBackground.lineStyle.join .....	3138
	Erlaubte Werte .....	3139
6.7.105	chart.dataZoom.selectedDataBackground.lineStyle.miterLimit.....	3139
6.7.106	chart.dataZoom.selectedDataBackground.lineStyle.shadowBlur .....	3139
6.7.107	chart.dataZoom.selectedDataBackground.lineStyle.shadowColor .....	3140
6.7.108	chart.dataZoom.selectedDataBackground.lineStyle.shadowOffsetX .....	3140
6.7.109	chart.dataZoom.selectedDataBackground.lineStyle.shadowOffsetY .....	3140
6.7.110	chart.dataZoom.selectedDataBackground.lineStyle.opacity.....	3141
6.7.111	chart.dataZoom.selectedDataBackground.areaStyle.object .....	3141
6.7.112	chart.dataZoom.selectedDataBackground.areaStyle.color .....	3141
6.7.113	chart.dataZoom.selectedDataBackground.areaStyle.shadowBlur .....	3143
6.7.114	chart.dataZoom.selectedDataBackground.areaStyle.shadowColor .....	3144
6.7.115	chart.dataZoom.selectedDataBackground.areaStyle.shadowOffsetX.....	3144
6.7.116	chart.dataZoom.selectedDataBackground.areaStyle.shadowOffsetY .....	3145
6.7.117	chart.dataZoom.selectedDataBackground.areaStyle.opacity.....	3145
6.7.118	chart.dataZoom.moveHandleIcon.....	3145
6.7.119	chart.dataZoom.moveHandleSize .....	3146
6.7.120	chart.dataZoom.moveHandleStyle.object.....	3147
6.7.121	chart.dataZoom.moveHandleStyle.color .....	3147
6.7.122	chart.dataZoom.moveHandleStyle.borderColor .....	3148
6.7.123	chart.dataZoom.moveHandleStyle.borderWidth .....	3149
6.7.124	chart.dataZoom.moveHandleStyle.borderType.....	3149
	Erlaubte Werte .....	3149
6.7.125	chart.dataZoom.moveHandleStyle.borderDashOffset.....	3150
6.7.126	chart.dataZoom.moveHandleStyle.borderCap .....	3150
	Erlaubte Werte .....	3150
6.7.127	chart.dataZoom.moveHandleStyle.borderJoin.....	3150
	Erlaubte Werte .....	3151
6.7.128	chart.dataZoom.moveHandleStyle.borderMiterLimit.....	3151
6.7.129	chart.dataZoom.moveHandleStyle.shadowBlur .....	3152
6.7.130	chart.dataZoom.moveHandleStyle.shadowColor .....	3152

6.7.131	chart.dataZoom.moveHandleStyle.shadowOffsetX .....	3152
6.7.132	chart.dataZoom.moveHandleStyle.shadowOffsetY .....	3153
6.7.133	chart.dataZoom.moveHandleStyle.opacity .....	3153
6.7.134	chart.dataZoom.brushSelect .....	3153
6.7.135	chart.dataZoom.brushStyle.object.....	3154
6.7.136	chart.dataZoom.brushStyle.color.....	3154
6.7.137	chart.dataZoom.brushStyle.borderColor.....	3155
6.7.138	chart.dataZoom.brushStyle.borderWidth.....	3155
6.7.139	chart.dataZoom.brushStyle.borderType.....	3155
	Erlaubte Werte .....	3156
6.7.140	chart.dataZoom.brushStyle.borderDashOffset .....	3156
6.7.141	chart.dataZoom.brushStyle.borderCap .....	3156
	Erlaubte Werte .....	3157
6.7.142	chart.dataZoom.brushStyle.borderJoin .....	3157
	Erlaubte Werte .....	3157
6.7.143	chart.dataZoom.brushStyle.borderMiterLimit .....	3157
6.7.144	chart.dataZoom.brushStyle.shadowBlur .....	3158
6.7.145	chart.dataZoom.brushStyle.shadowColor .....	3158
6.7.146	chart.dataZoom.brushStyle.shadowOffsetX.....	3159
6.7.147	chart.dataZoom.brushStyle.shadowOffsetY.....	3159
6.7.148	chart.dataZoom.brushStyle.opacity.....	3159
<b>6.8</b>	<b>ECharts (5) Gantt .....</b>	<b>3160</b>
6.8.1	chart.gantt.column.categoryname.....	3160
6.8.2	chart.gantt.column.categorygroup .....	3160
6.8.3	chart.gantt.column.barinfo.....	3161
6.8.4	chart.gantt.column.iconinfo .....	3161
6.8.5	chart.gantt.column.barcolor.....	3162
6.8.6	chart.gantt.screenitems .....	3162
6.8.7	chart.gantt.screenitems.height.....	3163
6.8.8	chart.gantt.iconlink.check.....	3163
6.8.9	chart.gantt.iconlink.path.....	3164
6.8.10	chart.gantt.icon.color .....	3164
6.8.11	chart.gantt.nodata.text .....	3165
6.8.12	chart.gantt.nodata.color .....	3165
6.8.13	chart.gantt.nodata.textColor .....	3166
6.8.14	chart.gantt.grid.height .....	3166

6.8.15	chart.gantt.grid.left.....	3167
6.8.16	chart.gantt.grid.top .....	3167
6.8.17	chart.gantt.grid.right .....	3167
6.8.18	chart.gantt.grid.bottom .....	3168
6.8.19	chart.gantt.nodata.maskColor.....	3168
6.8.20	chart.gantt.nodata.zlevel .....	3169
6.8.21	chart.gantt.nodata.fontSize .....	3169
6.8.22	chart.gantt.nodata.showSpinner.....	3170
6.8.23	chart.gantt.nodata.spinnerRadius.....	3170
6.8.24	chart.gantt.nodata.lineWidth.....	3171
6.8.25	chart.gantt.axis.path .....	3171
6.8.26	chart.gantt.axis.x .....	3172
6.8.27	chart.gantt.axis.y .....	3172
6.8.28	chart.gantt.axis.width.....	3173
6.8.29	chart.gantt.axis.height.....	3173
6.8.30	chart.gantt.axis.layout.....	3174
	Erlaubte Werte .....	3174
6.8.31	chart.gantt.axis.style.fill .....	3174
6.8.32	chart.gantt.axis.categorynametext.style.x.....	3175
6.8.33	chart.gantt.axis.categorynametext.style.y.....	3175
6.8.34	chart.gantt.axis.categorynametext.style.textVerticalAlign .....	3176
	Erlaubte Werte .....	3176
6.8.35	chart.gantt.axis.categorynametext.style.textAlign.....	3176
	Erlaubte Werte .....	3177
6.8.36	chart.gantt.axis.categorynametext.style.textFill .....	3177
6.8.37	chart.gantt.axis.categorygrouptext.style.x .....	3177
6.8.38	chart.gantt.axis.categorygrouptext.style.y .....	3178
6.8.39	chart.gantt.axis.categorygrouptext.style.textVerticalAlign.....	3178
	Erlaubte Werte .....	3179
6.8.40	chart.gantt.axis.categorygrouptext.style.textAlign .....	3179
	Erlaubte Werte .....	3179
6.8.41	chart.gantt.axis.categorygrouptext.style.textFill.....	3180
6.8.42	chart.gantt.rect.style.textFill.....	3180
6.8.43	chart.gantt.rect.style.stroke.....	3181
6.8.44	chart.gantt.rect.style.fill .....	3181
6.9	ECharts (5) General .....	3182

6.9.1	chart.type .....	3182
6.9.2	chart.theme .....	3182
<b>6.10</b>	<b>ECharts (5) Geo .....</b>	<b>3183</b>
6.10.1	chart.geo.object .....	3183
6.10.2	chart.geo.id .....	3183
6.10.3	chart.geo.show .....	3184
6.10.4	chart.geo.map .....	3184
6.10.5	chart.geo.roam .....	3185
	Erlaubte Werte .....	3186
6.10.6	chart.geo.center .....	3186
6.10.7	chart.geo.aspectScale .....	3186
6.10.8	chart.geo.boundingCoords .....	3187
6.10.9	chart.geo.zoom .....	3187
6.10.10	chart.geo.scaleLimit.object .....	3187
6.10.11	chart.geo.scaleLimit.min .....	3188
6.10.12	chart.geo.scaleLimit.max .....	3188
6.10.13	chart.geo.nameMap .....	3188
6.10.14	chart.geo.selectedMode .....	3189
6.10.15	chart.geo.label.object .....	3189
6.10.16	chart.geo.label.show .....	3190
6.10.17	chart.geo.label.position .....	3190
	Erlaubte Werte .....	3191
6.10.18	chart.geo.label.distance .....	3191
6.10.19	chart.geo.label.rotate .....	3191
6.10.20	chart.geo.label.offset .....	3192
6.10.21	chart.geo.label.formatter .....	3192
6.10.22	chart.geo.label.color .....	3193
6.10.23	chart.geo.label.fontStyle .....	3194
	Erlaubte Werte .....	3194
6.10.24	chart.geo.label.fontWeight .....	3194
	Erlaubte Werte .....	3195
6.10.25	chart.geo.label.fontFamily .....	3195
6.10.26	chart.geo.label.fontSize .....	3195
6.10.27	chart.geo.label.align .....	3195
	Erlaubte Werte .....	3196
6.10.28	chart.geo.label.verticalAlign .....	3196



	Erlaubte Werte .....	3197
6.10.29	chart.geo.label.lineHeight .....	3197
6.10.30	chart.geo.label.backgroundColor .....	3197
6.10.31	chart.geo.label.borderColor .....	3198
6.10.32	chart.geo.label.borderWidth .....	3199
6.10.33	chart.geo.label.borderRadius .....	3199
6.10.34	chart.geo.label.padding .....	3199
6.10.35	chart.geo.label.shadowColor .....	3200
6.10.36	chart.geo.label.shadowBlur .....	3200
6.10.37	chart.geo.label.shadowOffsetX .....	3200
6.10.38	chart.geo.label.shadowOffsetY .....	3201
6.10.39	chart.geo.label.width .....	3201
6.10.40	chart.geo.label.height .....	3202
6.10.41	chart.geo.label.textBorderColor .....	3202
6.10.42	chart.geo.label.textBorderWidth .....	3202
6.10.43	chart.geo.label.textShadowColor .....	3203
6.10.44	chart.geo.label.textShadowBlur .....	3203
6.10.45	chart.geo.label.textShadowOffsetX .....	3204
6.10.46	chart.geo.label.textShadowOffsetY .....	3204
6.10.47	chart.geo.label.rich.object .....	3204
6.10.48	chart.geo.label.emphasis.object .....	3205
6.10.49	chart.geo.label.emphasis.show .....	3206
6.10.50	chart.geo.label.emphasis.position .....	3206
	Erlaubte Werte .....	3207
6.10.51	chart.geo.label.emphasis.distance .....	3207
6.10.52	chart.geo.label.emphasis.rotate .....	3207
6.10.53	chart.geo.label.emphasis.offset .....	3208
6.10.54	chart.geo.label.emphasis.formatter .....	3208
6.10.55	chart.geo.label.emphasis.color .....	3209
6.10.56	chart.geo.label.emphasis.fontStyle .....	3209
	Erlaubte Werte .....	3210
6.10.57	chart.geo.label.emphasis.fontWeight .....	3210
	Erlaubte Werte .....	3210
6.10.58	chart.geo.label.emphasis.fontFamily .....	3211
6.10.59	chart.geo.label.emphasis.fontSize .....	3211
6.10.60	chart.geo.label.emphasis.align .....	3211

	Erlaubte Werte .....	3212
6.10.61	chart.geo.label.emphasis.verticalAlign .....	3212
	Erlaubte Werte .....	3213
6.10.62	chart.geo.label.emphasis.lineHeight.....	3213
6.10.63	chart.geo.label.emphasis.backgroundColor.....	3213
6.10.64	chart.geo.label.emphasis.borderColor.....	3214
6.10.65	chart.geo.label.emphasis.borderWidth.....	3214
6.10.66	chart.geo.label.emphasis.borderRadius .....	3215
6.10.67	chart.geo.label.emphasis.padding .....	3215
6.10.68	chart.geo.label.emphasis.shadowColor .....	3215
6.10.69	chart.geo.label.emphasis.shadowBlur .....	3216
6.10.70	chart.geo.label.emphasis.shadowOffsetX.....	3216
6.10.71	chart.geo.label.emphasis.shadowOffsetY .....	3217
6.10.72	chart.geo.label.emphasis.width .....	3217
6.10.73	chart.geo.label.emphasis.height .....	3217
6.10.74	chart.geo.label.emphasis.textBorderColor .....	3218
6.10.75	chart.geo.label.emphasis.textBorderWidth .....	3218
6.10.76	chart.geo.label.emphasis.textShadowColor .....	3219
6.10.77	chart.geo.label.emphasis.textShadowBlur .....	3219
6.10.78	chart.geo.label.emphasis.textShadowOffsetX.....	3219
6.10.79	chart.geo.label.emphasis.textShadowOffsetY .....	3220
6.10.80	chart.geo.label.emphasis.rich.object .....	3220
6.10.81	chart.geo.itemStyle.object.....	3221
6.10.82	chart.geo.itemStyle.areaColor.object .....	3222
6.10.83	chart.geo.itemStyle.areaColor.color .....	3222
6.10.84	chart.geo.itemStyle.areaColor.borderColor .....	3223
6.10.85	chart.geo.itemStyle.areaColor.borderWidth.....	3224
6.10.86	chart.geo.itemStyle.areaColor.borderType .....	3224
	Erlaubte Werte .....	3224
6.10.87	chart.geo.itemStyle.areaColor.shadowBlur.....	3225
6.10.88	chart.geo.itemStyle.areaColor.shadowColor.....	3225
6.10.89	chart.geo.itemStyle.areaColor.shadowOffsetX.....	3225
6.10.90	chart.geo.itemStyle.areaColor.shadowOffsetY .....	3226
6.10.91	chart.geo.itemStyle.areaColor.opacity .....	3226
6.10.92	chart.geo.itemStyle.emphasis.object.....	3226
6.10.93	chart.geo.itemStyle.emphasis.areaColor .....	3227

6.10.94	chart.geo.itemStyle.emphasis.color .....	3227
6.10.95	chart.geo.itemStyle.emphasis.borderColor .....	3228
6.10.96	chart.geo.itemStyle.emphasis.borderWidth .....	3229
6.10.97	chart.geo.itemStyle.emphasis.borderType .....	3229
	Erlaubte Werte .....	3229
6.10.98	chart.geo.itemStyle.emphasis.shadowBlur .....	3230
6.10.99	chart.geo.itemStyle.emphasis.shadowColor .....	3230
6.10.100	chart.geo.itemStyle.emphasis.shadowOffsetX .....	3230
6.10.101	chart.geo.itemStyle.emphasis.shadowOffsetY .....	3231
6.10.102	chart.geo.itemStyle.emphasis.opacity .....	3231
6.10.103	chart.geo.zlevel .....	3231
6.10.104	chart.geo.z .....	3232
6.10.105	chart.geo.left .....	3232
	Erlaubte Werte .....	3233
6.10.106	chart.geo.top .....	3233
	Erlaubte Werte .....	3233
6.10.107	chart.geo.right .....	3234
	Erlaubte Werte .....	3234
6.10.108	chart.geo.bottom .....	3234
	Erlaubte Werte .....	3235
6.10.109	chart.geo.layoutCenter .....	3235
6.10.110	chart.geo.layoutSize .....	3235
6.10.111	chart.geo.regions.{x}.object .....	3236
6.10.112	chart.geo.regions.{x}.name .....	3236
6.10.113	chart.geo.regions.{x}.selected .....	3236
6.10.114	chart.geo.regions.{x}.itemStyle.object .....	3237
6.10.115	chart.geo.regions.{x}.itemStyle.areaColor .....	3237
6.10.116	chart.geo.regions.{x}.itemStyle.color .....	3238
6.10.117	chart.geo.regions.{x}.itemStyle.borderColor .....	3239
6.10.118	chart.geo.regions.{x}.itemStyle.borderWidth .....	3240
6.10.119	chart.geo.regions.{x}.itemStyle.borderType .....	3240
	Erlaubte Werte .....	3240
6.10.120	chart.geo.regions.{x}.itemStyle.shadowBlur .....	3241
6.10.121	chart.geo.regions.{x}.itemStyle.shadowColor .....	3241
6.10.122	chart.geo.regions.{x}.itemStyle.shadowOffsetX .....	3241
6.10.123	chart.geo.regions.{x}.itemStyle.shadowOffsetY .....	3242

6.10.124	chart.geo.regions.{x}.itemStyle.opacity .....	3242
6.10.125	chart.geo.regions.{x}.itemStyle.emphasis.object .....	3242
6.10.126	chart.geo.regions.{x}.itemStyle.emphasis.areaColor .....	3243
6.10.127	chart.geo.regions.{x}.itemStyle.emphasis.color .....	3243
6.10.128	chart.geo.regions.{x}.itemStyle.emphasis.borderColor .....	3244
6.10.129	chart.geo.regions.{x}.itemStyle.emphasis.borderWidth .....	3245
6.10.130	chart.geo.regions.{x}.itemStyle.emphasis.borderType .....	3245
	Erlaubte Werte .....	3245
6.10.131	chart.geo.regions.{x}.itemStyle.emphasis.shadowBlur.....	3246
6.10.132	chart.geo.regions.{x}.itemStyle.emphasis.shadowColor .....	3246
6.10.133	chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetX .....	3246
6.10.134	chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetY.....	3247
6.10.135	chart.geo.regions.{x}.itemStyle.emphasis.opacity .....	3247
6.10.136	chart.geo.regions.{x}.label.object .....	3247
6.10.137	chart.geo.regions.{x}.label.show.....	3248
6.10.138	chart.geo.regions.{x}.label.position .....	3248
	Erlaubte Werte .....	3249
6.10.139	chart.geo.regions.{x}.label.distance .....	3249
6.10.140	chart.geo.regions.{x}.label.rotate .....	3249
6.10.141	chart.geo.regions.{x}.label.offset .....	3250
6.10.142	chart.geo.regions.{x}.label.formatter .....	3250
6.10.143	chart.geo.regions.{x}.label.color .....	3251
6.10.144	chart.geo.regions.{x}.label.fontStyle .....	3252
	Erlaubte Werte .....	3252
6.10.145	chart.geo.regions.{x}.label.fontWeight .....	3252
	Erlaubte Werte .....	3253
6.10.146	chart.geo.regions.{x}.label.fontFamily .....	3253
6.10.147	chart.geo.regions.{x}.label.fontSize.....	3253
6.10.148	chart.geo.regions.{x}.label.align .....	3254
	Erlaubte Werte .....	3254
6.10.149	chart.geo.regions.{x}.label.verticalAlign.....	3254
	Erlaubte Werte .....	3255
6.10.150	chart.geo.regions.{x}.label.lineHeight .....	3255
6.10.151	chart.geo.regions.{x}.label.backgroundColor .....	3255
6.10.152	chart.geo.regions.{x}.label.borderColor .....	3256
6.10.153	chart.geo.regions.{x}.label.borderWidth .....	3257

6.10.154	chart.geo.regions.{x}.label.borderRadius.....	3257
6.10.155	chart.geo.regions.{x}.label.padding.....	3257
6.10.156	chart.geo.regions.{x}.label.shadowColor .....	3258
6.10.157	chart.geo.regions.{x}.label.shadowBlur .....	3258
6.10.158	chart.geo.regions.{x}.label.shadowOffsetX .....	3258
6.10.159	chart.geo.regions.{x}.label.shadowOffsetY .....	3259
6.10.160	chart.geo.regions.{x}.label.width.....	3259
6.10.161	chart.geo.regions.{x}.label.height.....	3260
6.10.162	chart.geo.regions.{x}.label.textBorderColor .....	3260
6.10.163	chart.geo.regions.{x}.label.textBorderWidth.....	3260
6.10.164	chart.geo.regions.{x}.label.textShadowColor .....	3261
6.10.165	chart.geo.regions.{x}.label.textShadowBlur .....	3261
6.10.166	chart.geo.regions.{x}.label.textShadowOffsetX .....	3262
6.10.167	chart.geo.regions.{x}.label.textShadowOffsetY .....	3262
6.10.168	chart.geo.regions.{x}.label.rich.object.....	3262
6.10.169	chart.geo.regions.{x}.label.emphasis.object.....	3263
6.10.170	chart.geo.regions.{x}.label.emphasis.show.....	3264
6.10.171	chart.geo.regions.{x}.label.emphasis.position.....	3264
	Erlaubte Werte .....	3265
6.10.172	chart.geo.regions.{x}.label.emphasis.distance .....	3265
6.10.173	chart.geo.regions.{x}.label.emphasis.rotate .....	3265
6.10.174	chart.geo.regions.{x}.label.emphasis.offset.....	3266
6.10.175	chart.geo.regions.{x}.label.emphasis.formatter .....	3266
6.10.176	chart.geo.regions.{x}.label.emphasis.color .....	3267
6.10.177	chart.geo.regions.{x}.label.emphasis.fontStyle .....	3267
	Erlaubte Werte .....	3268
6.10.178	chart.geo.regions.{x}.label.emphasis.fontWeight.....	3268
	Erlaubte Werte .....	3268
6.10.179	chart.geo.regions.{x}.label.emphasis.fontFamily .....	3269
6.10.180	chart.geo.regions.{x}.label.emphasis.fontSize.....	3269
6.10.181	chart.geo.regions.{x}.label.emphasis.align .....	3269
	Erlaubte Werte .....	3270
6.10.182	chart.geo.regions.{x}.label.emphasis.verticalAlign.....	3270
	Erlaubte Werte .....	3271
6.10.183	chart.geo.regions.{x}.label.emphasis.lineHeight .....	3271
6.10.184	chart.geo.regions.{x}.label.emphasis.backgroundColor .....	3271

6.10.185	chart.geo.regions.{x}.label.emphasis.borderColor .....	3272
6.10.186	chart.geo.regions.{x}.label.emphasis.borderWidth .....	3272
6.10.187	chart.geo.regions.{x}.label.emphasis.borderRadius.....	3273
6.10.188	chart.geo.regions.{x}.label.emphasis.padding.....	3273
6.10.189	chart.geo.regions.{x}.label.emphasis.shadowColor .....	3273
6.10.190	chart.geo.regions.{x}.label.emphasis.shadowBlur .....	3274
6.10.191	chart.geo.regions.{x}.label.emphasis.shadowOffsetX .....	3274
6.10.192	chart.geo.regions.{x}.label.emphasis.shadowOffsetY .....	3275
6.10.193	chart.geo.regions.{x}.label.emphasis.width.....	3275
6.10.194	chart.geo.regions.{x}.label.emphasis.height.....	3275
6.10.195	chart.geo.regions.{x}.label.emphasis.textBorderColor .....	3276
6.10.196	chart.geo.regions.{x}.label.emphasis.textBorderWidth .....	3276
6.10.197	chart.geo.regions.{x}.label.emphasis.textShadowColor .....	3277
6.10.198	chart.geo.regions.{x}.label.emphasis.textShadowBlur .....	3277
6.10.199	chart.geo.regions.{x}.label.emphasis.textShadowOffsetX .....	3277
6.10.200	chart.geo.regions.{x}.label.emphasis.textShadowOffsetY .....	3278
6.10.201	chart.geo.regions.{x}.label.emphasis.rich.object .....	3278
6.10.202	chart.geo.silent .....	3279
6.11	ECharts (5) Graphic .....	3280
6.11.1	chart.graphic.object .....	3280
6.12	ECharts (5) Grid .....	3287
6.12.1	chart.grid.object.....	3287
6.12.2	chart.grid.{x}.id.....	3287
6.12.3	chart.grid.{x}.show .....	3288
6.12.4	chart.grid.{x}.zlevel .....	3288
6.12.5	chart.grid.{x}.z .....	3288
6.12.6	chart.grid.{x}.left .....	3289
	Erlaubte Werte .....	3289
6.12.7	chart.grid.{x}.top .....	3289
	Erlaubte Werte .....	3290
6.12.8	chart.grid.{x}.right.....	3290
	Erlaubte Werte .....	3290
6.12.9	chart.grid.{x}.bottom .....	3291
	Erlaubte Werte .....	3291
6.12.10	chart.grid.{x}.width .....	3291

6.12.11	chart.grid.{x}.height .....	3292
6.12.12	chart.grid.{x}.containLabel .....	3292
6.12.13	chart.grid.{x}.backgroundColor.....	3293
6.12.14	chart.grid.{x}.borderColor .....	3293
6.12.15	chart.grid.{x}.borderWidth.....	3293
6.12.16	chart.grid.{x}.shadowBlur.....	3294
6.12.17	chart.grid.{x}.shadowColor.....	3294
6.12.18	chart.grid.{x}.shadowOffsetX.....	3295
6.12.19	chart.grid.{x}.shadowOffsetY .....	3295
6.12.20	chart.grid.{x}.tooltip.object .....	3295
6.12.21	chart.grid.{x}.tooltip.show .....	3296
6.12.22	chart.grid.{x}.tooltip.trigger .....	3296
	Erlaubte Werte .....	3297
6.12.23	chart.grid.{x}.tooltip.axisPointer.object .....	3297
6.12.24	chart.grid.{x}.tooltip.axisPointer.type .....	3297
	Erlaubte Werte .....	3298
6.12.25	chart.grid.{x}.tooltip.axisPointer.axis .....	3298
	Erlaubte Werte .....	3298
6.12.26	chart.grid.{x}.tooltip.axisPointer.snap.....	3299
6.12.27	chart.grid.{x}.tooltip.axisPointer.z.....	3299
6.12.28	chart.grid.{x}.tooltip.axisPointer.label.object.....	3299
6.12.29	chart.grid.{x}.tooltip.axisPointer.label.show .....	3300
6.12.30	chart.grid.{x}.tooltip.axisPointer.label.precision .....	3300
6.12.31	chart.grid.{x}.tooltip.axisPointer.label.formatter .....	3300
6.12.32	chart.grid.{x}.tooltip.axisPointer.label.margin .....	3302
6.12.33	chart.grid.{x}.tooltip.axisPointer.label.color.....	3302
6.12.34	chart.grid.{x}.tooltip.axisPointer.label.fontStyle .....	3303
6.12.35	chart.grid.{x}.tooltip.axisPointer.label.fontWeight.....	3303
	Erlaubte Werte .....	3304
6.12.36	chart.grid.{x}.tooltip.axisPointer.label.fontFamily .....	3304
6.12.37	chart.grid.{x}.tooltip.axisPointer.label.fontSize.....	3304
6.12.38	chart.grid.{x}.tooltip.axisPointer.label.lineHeight .....	3304
6.12.39	chart.grid.{x}.tooltip.axisPointer.label.width.....	3305
6.12.40	chart.grid.{x}.tooltip.axisPointer.label.height.....	3306
6.12.41	chart.grid.{x}.tooltip.axisPointer.label.textBorderColor .....	3306
6.12.42	chart.grid.{x}.tooltip.axisPointer.label.textBorderWidth .....	3306

6.12.43	chart.grid.{x}.tooltip.axisPointer.label.textShadowColor .....	3307
6.12.44	chart.grid.{x}.tooltip.axisPointer.label.textShadowBlur .....	3307
6.12.45	chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetX .....	3307
6.12.46	chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetY .....	3308
6.12.47	chart.grid.{x}.tooltip.axisPointer.label.padding .....	3308
6.12.48	chart.grid.{x}.tooltip.axisPointer.label.backgroundColor .....	3309
6.12.49	chart.grid.{x}.tooltip.axisPointer.label.borderColor .....	3309
6.12.50	chart.grid.{x}.tooltip.axisPointer.label.borderWidth .....	3310
6.12.51	chart.grid.{x}.tooltip.axisPointer.label.shadowBlur .....	3310
6.12.52	chart.grid.{x}.tooltip.axisPointer.label.shadowColor .....	3310
6.12.53	chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetX .....	3311
6.12.54	chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetY .....	3311
6.12.55	chart.grid.{x}.tooltip.axisPointer.lineStyle.object.....	3312
6.12.56	chart.grid.{x}.tooltip.axisPointer.lineStyle.color .....	3312
6.12.57	chart.grid.{x}.tooltip.axisPointer.lineStyle.width .....	3313
6.12.58	chart.grid.{x}.tooltip.axisPointer.lineStyle.type.....	3314
	Erlaubte Werte .....	3314
6.12.59	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowBlur .....	3314
6.12.60	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowColor .....	3315
6.12.61	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetX.....	3315
6.12.62	chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetY .....	3315
6.12.63	chart.grid.{x}.tooltip.axisPointer.lineStyle.opacity.....	3316
6.12.64	chart.grid.{x}.tooltip.axisPointer.shadowStyle.object.....	3316
6.12.65	chart.grid.{x}.tooltip.axisPointer.shadowStyle.color .....	3316
6.12.66	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowBlur .....	3318
6.12.67	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowColor .....	3319
6.12.68	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetX.....	3319
6.12.69	chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	3320
6.12.70	chart.grid.{x}.tooltip.axisPointer.shadowStyle.opacity.....	3320
6.12.71	chart.grid.{x}.tooltip.axisPointer.crossStyle.object.....	3320
6.12.72	chart.grid.{x}.tooltip.axisPointer.crossStyle.color .....	3321
6.12.73	chart.grid.{x}.tooltip.axisPointer.crossStyle.width .....	3322
6.12.74	chart.grid.{x}.tooltip.axisPointer.crossStyle.type .....	3323
	Erlaubte Werte .....	3323
6.12.75	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowBlur .....	3323
6.12.76	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowColor .....	3324



6.12.77	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetX .....	3324
6.12.78	chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetY .....	3324
6.12.79	chart.grid.{x}.tooltip.axisPointer.crossStyle.opacity .....	3325
6.12.80	chart.grid.{x}.tooltip.position .....	3325
6.12.81	chart.grid.{x}.tooltip.formatter .....	3327
6.12.82	chart.grid.{x}.tooltip.backgroundColor .....	3329
6.12.83	chart.grid.{x}.tooltip.borderColor .....	3329
6.12.84	chart.grid.{x}.tooltip.borderWidth .....	3330
6.12.85	chart.grid.{x}.tooltip.padding.....	3330
6.12.86	chart.grid.{x}.tooltip.textStyle.object .....	3331
6.12.87	chart.grid.{x}.tooltip.textStyle.color .....	3331
6.12.88	chart.grid.{x}.tooltip.textStyle.fontStyle .....	3331
	Erlaubte Werte .....	3332
6.12.89	chart.grid.{x}.tooltip.textStyle.fontWeight .....	3332
	Erlaubte Werte .....	3332
6.12.90	chart.grid.{x}.tooltip.textStyle.fontFamily.....	3332
6.12.91	chart.grid.{x}.tooltip.textStyle.fontSize .....	3333
6.12.92	chart.grid.{x}.tooltip.textStyle.lineHeight .....	3333
6.12.93	chart.grid.{x}.tooltip.textStyle.width .....	3334
6.12.94	chart.grid.{x}.tooltip.textStyle.height.....	3334
6.12.95	chart.grid.{x}.tooltip.textStyle.textBorderColor.....	3335
6.12.96	chart.grid.{x}.tooltip.textStyle.textBorderWidth.....	3335
6.12.97	chart.grid.{x}.tooltip.textStyle.textShadowColor .....	3335
6.12.98	chart.grid.{x}.tooltip.textStyle.textShadowBlur.....	3336
6.12.99	chart.grid.{x}.tooltip.textStyle.textShadowOffsetX .....	3336
6.12.100	chart.grid.{x}.tooltip.textStyle.textShadowOffsetY.....	3336
6.12.101	chart.grid.{x}.tooltip.extraCssText .....	3337
6.12.102	chart.grid.{x}.tooltip.axisPointer.label.ellipsis.....	3337
6.12.103	chart.grid.{x}.tooltip.axisPointer.label.overflow.....	3338
	Erlaubte Werte .....	3338
6.12.104	chart.grid.{x}.tooltip.textStyle.ellipsis .....	3338
6.12.105	chart.grid.{x}.tooltip.textStyle.overflow .....	3338
	Erlaubte Werte .....	3339
6.12.106	chart.grid.{x}.tooltip.axisPointer.animation .....	3339
6.12.107	chart.grid.{x}.tooltip.axisPointer.animationThreshold .....	3339
6.12.108	chart.grid.{x}.tooltip.axisPointer.animationDuration.....	3340

6.12.109	chart.grid.{x}.tooltip.axisPointer.animationEasing .....	3340
	Erlaubte Werte .....	3341
6.12.110	chart.grid.{x}.tooltip.axisPointer.animationDelay .....	3341
6.12.111	chart.grid.{x}.tooltip.axisPointer.animationDurationUpdate .....	3341
6.12.112	chart.grid.{x}.tooltip.axisPointer.animationEasingUpdate .....	3342
	Erlaubte Werte .....	3342
6.12.113	chart.grid.{x}.tooltip.axisPointer.animationDelayUpdate .....	3342
<b>6.13</b>	<b>ECharts (5) Legend .....</b>	<b>3343</b>
6.13.1	chart.legend.object .....	3343
6.13.2	chart.legend.type .....	3343
	Erlaubte Werte .....	3344
6.13.3	chart.legend.id .....	3344
6.13.4	chart.legend.show .....	3345
6.13.5	chart.legend.zlevel .....	3345
6.13.6	chart.legend.z .....	3345
6.13.7	chart.legend.left .....	3346
	Erlaubte Werte .....	3346
6.13.8	chart.legend.top .....	3346
	Erlaubte Werte .....	3347
6.13.9	chart.legend.right .....	3347
6.13.10	chart.legend.bottom .....	3347
	Erlaubte Werte .....	3348
6.13.11	chart.legend.width .....	3348
6.13.12	chart.legend.height .....	3348
6.13.13	chart.legend.orient .....	3349
	Erlaubte Werte .....	3349
6.13.14	chart.legend.align .....	3349
	Erlaubte Werte .....	3350
6.13.15	chart.legend.padding .....	3350
6.13.16	chart.legend.itemGap .....	3350
6.13.17	chart.legend.itemWidth .....	3351
6.13.18	chart.legend.itemHeight .....	3351
6.13.19	chart.legend.symbolKeepAspect .....	3351
6.13.20	chart.legend.formatter .....	3352
6.13.21	chart.legend.selectedMode .....	3352
6.13.22	chart.legend.inactiveColor .....	3353

6.13.23	chart.legend.selected .....	3353
6.13.24	chart.legend.textStyle.object .....	3354
6.13.25	chart.legend.textStyle.color .....	3354
6.13.26	chart.legend.textStyle.fontStyle .....	3354
	Erlaubte Werte .....	3355
6.13.27	chart.legend.textStyle.fontWeight .....	3355
6.13.28	chart.legend.textStyle.fontFamily .....	3356
6.13.29	chart.legend.textStyle.fontSize .....	3356
6.13.30	chart.legend.textStyle.align .....	3356
	Erlaubte Werte .....	3357
6.13.31	chart.legend.textStyle.verticalAlign .....	3357
	Erlaubte Werte .....	3358
6.13.32	chart.legend.textStyle.lineHeight .....	3358
6.13.33	chart.legend.textStyle.backgroundColor .....	3358
6.13.34	chart.legend.textStyle.borderColor .....	3359
6.13.35	chart.legend.textStyle.borderWidth .....	3359
6.13.36	chart.legend.textStyle.borderRadius .....	3360
6.13.37	chart.legend.textStyle.padding .....	3360
6.13.38	chart.legend.textStyle.shadowColor .....	3361
6.13.39	chart.legend.textStyle.shadowBlur .....	3361
6.13.40	chart.legend.textStyle.shadowOffsetX .....	3361
6.13.41	chart.legend.textStyle.shadowOffsetY .....	3362
6.13.42	chart.legend.textStyle.width .....	3362
	Erlaubte Werte .....	3363
6.13.43	chart.legend.textStyle.height .....	3363
6.13.44	chart.legend.textStyle.textBorderColor .....	3363
6.13.45	chart.legend.textStyle.textBorderWidth .....	3363
6.13.46	chart.legend.textStyle.textShadowColor .....	3364
6.13.47	chart.legend.textStyle.textShadowBlur .....	3364
6.13.48	chart.legend.textStyle.textShadowOffsetX .....	3364
6.13.49	chart.legend.textStyle.textShadowOffsetY .....	3365
6.13.50	chart.legend.textStyle.rich .....	3365
6.13.51	chart.legend.tooltip .....	3366
6.13.52	chart.legend.data.object .....	3367
6.13.53	chart.legend.backgroundColor .....	3367
6.13.54	chart.legend.borderColor .....	3368

6.13.55	chart.legend.borderWidth .....	3368
6.13.56	chart.legend.borderRadius .....	3369
6.13.57	chart.legend.shadowBlur .....	3369
6.13.58	chart.legend.shadowColor .....	3369
6.13.59	chart.legend.shadowOffsetX .....	3370
6.13.60	chart.legend.shadowOffsetY .....	3370
6.13.61	chart.legend.scrollDataIndex .....	3371
6.13.62	chart.legend.pageButtonItemGap .....	3371
6.13.63	chart.legend.pageButtonGap .....	3371
6.13.64	chart.legend.pageButtonPosition .....	3372
6.13.65	chart.legend.pageFormatter .....	3372
6.13.66	chart.legend.pageIcons .....	3373
6.13.67	chart.legend.pageIcons.horizontal .....	3373
6.13.68	chart.legend.pageIcons.vertical .....	3374
6.13.69	chart.legend.pageIconColor .....	3374
6.13.70	chart.legend.pageIconInactiveColor .....	3375
6.13.71	chart.legend.pageIconSize .....	3375
6.13.72	chart.legend.pageTextStyle.object .....	3375
6.13.73	chart.legend.pageTextStyle.color .....	3376
6.13.74	chart.legend.pageTextStyle.fontStyle .....	3376
	Erlaubte Werte .....	3377
6.13.75	chart.legend.pageTextStyle.fontWeight .....	3377
6.13.76	chart.legend.pageTextStyle.fontFamily .....	3377
6.13.77	chart.legend.pageTextStyle.fontSize .....	3378
6.13.78	chart.legend.pageTextStyle.lineHeight .....	3378
6.13.79	chart.legend.pageTextStyle.width .....	3378
	Erlaubte Werte .....	3379
6.13.80	chart.legend.pageTextStyle.height .....	3379
6.13.81	chart.legend.pageTextStyle.textBorderColor .....	3379
6.13.82	chart.legend.pageTextStyle.textBorderWidth .....	3380
6.13.83	chart.legend.pageTextStyle.textShadowColor .....	3380
6.13.84	chart.legend.pageTextStyle.textShadowBlur .....	3381
6.13.85	chart.legend.pageTextStyle.textShadowOffsetX .....	3381
6.13.86	chart.legend.pageTextStyle.textShadowOffsetY .....	3381
6.13.87	chart.legend.animation .....	3382
6.13.88	chart.legend.animationDurationUpdate .....	3382

6.13.89	chart.legend.textStyle.overflow.....	3382
	Erlaubte Werte .....	3383
6.13.90	chart.legend.textStyle.ellipsis.....	3383
6.13.91	chart.legend.icon .....	3383
6.13.92	chart.legend.pageTextStyle.overflow.....	3384
	Erlaubte Werte .....	3384
6.13.93	chart.legend.pageTextStyle.ellipsis.....	3384
6.13.94	chart.legend.selector.....	3384
6.13.95	chart.legend.selectorPosition .....	3385
	Erlaubte Werte .....	3386
6.13.96	chart.legend.selectorItemGap .....	3386
6.13.97	chart.legend.selectorButtonGap .....	3386
6.13.98	chart.legend.selectorLabel.object.....	3386
6.13.99	chart.legend.selectorLabel.show.....	3387
6.13.100	chart.legend.selectorLabel.distance .....	3387
6.13.101	chart.legend.selectorLabel.rotate .....	3387
6.13.102	chart.legend.selectorLabel.offset .....	3388
6.13.103	chart.legend.selectorLabel.color.....	3388
6.13.104	chart.legend.selectorLabel.fontStyle .....	3389
	Erlaubte Werte .....	3389
6.13.105	chart.legend.selectorLabel.fontWeight.....	3389
	Erlaubte Werte .....	3389
6.13.106	chart.legend.selectorLabel.fontFamily .....	3390
6.13.107	chart.legend.selectorLabel.fontSize.....	3390
6.13.108	chart.legend.selectorLabel.align .....	3390
	Erlaubte Werte .....	3391
6.13.109	chart.legend.selectorLabel.verticalAlign.....	3391
	Erlaubte Werte .....	3392
6.13.110	chart.legend.selectorLabel.lineHeight .....	3392
6.13.111	chart.legend.selectorLabel.backgroundColor .....	3392
6.13.112	chart.legend.selectorLabel.borderColor .....	3393
6.13.113	chart.legend.selectorLabel.borderWidth .....	3394
6.13.114	chart.legend.selectorLabel.borderRadius.....	3394
6.13.115	chart.legend.selectorLabel.padding.....	3394
6.13.116	chart.legend.selectorLabel.shadowColor .....	3395
6.13.117	chart.legend.selectorLabel.shadowBlur .....	3395

6.13.118	chart.legend.selectorLabel.shadowOffsetX .....	3395
6.13.119	chart.legend.selectorLabel.shadowOffsetY .....	3396
6.13.120	chart.legend.selectorLabel.width .....	3396
6.13.121	chart.legend.selectorLabel.height .....	3396
6.13.122	chart.legend.selectorLabel.textBorderColor .....	3397
6.13.123	chart.legend.selectorLabel.textBorderWidth .....	3397
6.13.124	chart.legend.selectorLabel.textShadowColor .....	3398
6.13.125	chart.legend.selectorLabel.textShadowBlur .....	3398
6.13.126	chart.legend.selectorLabel.textShadowOffsetX .....	3398
6.13.127	chart.legend.selectorLabel.textShadowOffsetY .....	3399
6.13.128	chart.legend.selectorLabel.overflow .....	3399
	Erlaubte Werte .....	3399
6.13.129	chart.legend.selectorLabel.ellipsis .....	3399
6.13.130	chart.legend.selectorLabel.rich.object .....	3400
<b>6.14</b>	<b>ECharts (5) Parallel .....</b>	<b>3401</b>
6.14.1	chart.parallel.object .....	3401
6.14.2	chart.parallel.id .....	3404
6.14.3	chart.parallel.zlevel .....	3405
6.14.4	chart.parallel.z .....	3405
6.14.5	chart.parallel.left .....	3406
	Erlaubte Werte .....	3406
6.14.6	chart.parallel.top .....	3406
	Erlaubte Werte .....	3407
6.14.7	chart.parallel.right .....	3407
	Erlaubte Werte .....	3407
6.14.8	chart.parallel.bottom .....	3407
6.14.9	chart.parallel.width .....	3408
6.14.10	chart.parallel.height .....	3408
6.14.11	chart.parallel.layout .....	3408
	Erlaubte Werte .....	3409
6.14.12	chart.parallel.axisExpandable .....	3409
6.14.13	chart.parallel.axisExpandCenter .....	3409
6.14.14	chart.parallel.axisExpandCount .....	3410
6.14.15	chart.parallel.axisExpandWidth .....	3410
6.14.16	chart.parallel.axisExpandTriggerOn .....	3411
	Erlaubte Werte .....	3411

6.14.17	chart.parallel.parallelAxisDefault.object.....	3411
6.14.18	chart.parallel.parallelAxisDefault.type.....	3411
	Erlaubte Werte .....	3412
6.14.19	chart.parallel.parallelAxisDefault.name.....	3412
6.14.20	chart.parallel.parallelAxisDefault.nameLocation.....	3412
	Erlaubte Werte .....	3413
6.14.21	chart.parallel.parallelAxisDefault.nameTextStyle.object.....	3413
6.14.22	chart.parallel.parallelAxisDefault.nameTextStyle.color.....	3413
6.14.23	chart.parallel.parallelAxisDefault.nameTextStyle.fontStyle.....	3414
	Erlaubte Werte .....	3414
6.14.24	chart.parallel.parallelAxisDefault.nameTextStyle.fontWeight.....	3414
	Erlaubte Werte .....	3414
6.14.25	chart.parallel.parallelAxisDefault.nameTextStyle.fontFamily.....	3415
6.14.26	chart.parallel.parallelAxisDefault.nameTextStyle.fontSize.....	3415
6.14.27	chart.parallel.parallelAxisDefault.nameTextStyle.align.....	3415
	Erlaubte Werte .....	3416
6.14.28	chart.parallel.parallelAxisDefault.nameTextStyle.verticalAlign.....	3416
	Erlaubte Werte .....	3417
6.14.29	chart.parallel.parallelAxisDefault.nameTextStyle.lineHeight.....	3417
6.14.30	chart.parallel.parallelAxisDefault.nameTextStyle.backgroundColor.....	3417
6.14.31	chart.parallel.parallelAxisDefault.nameTextStyle.borderColor.....	3418
6.14.32	chart.parallel.parallelAxisDefault.nameTextStyle.borderWidth.....	3418
6.14.33	chart.parallel.parallelAxisDefault.nameTextStyle.borderRadius.....	3419
6.14.34	chart.parallel.parallelAxisDefault.nameTextStyle.padding.....	3419
6.14.35	chart.parallel.parallelAxisDefault.nameTextStyle.shadowColor.....	3420
6.14.36	chart.parallel.parallelAxisDefault.nameTextStyle.shadowBlur.....	3420
6.14.37	chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetX.....	3420
6.14.38	chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetY.....	3421
6.14.39	chart.parallel.parallelAxisDefault.nameTextStyle.width.....	3421
6.14.40	chart.parallel.parallelAxisDefault.nameTextStyle.height.....	3422
6.14.41	chart.parallel.parallelAxisDefault.nameTextStyle.textBorderColor.....	3422
6.14.42	chart.parallel.parallelAxisDefault.nameTextStyle.textBorderWidth.....	3422
6.14.43	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowColor.....	3423
6.14.44	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowBlur.....	3423
6.14.45	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetX.....	3423
6.14.46	chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetY.....	3424

6.14.47	chart.parallel.parallelAxisDefault.nameTextStyle.rich.object .....	3424
6.14.48	chart.parallel.parallelAxisDefault.nameGap .....	3425
6.14.49	chart.parallel.parallelAxisDefault.nameRotate.....	3426
6.14.50	chart.parallel.parallelAxisDefault.inverse .....	3426
6.14.51	chart.parallel.parallelAxisDefault.boundaryGap .....	3426
6.14.52	chart.parallel.parallelAxisDefault.min .....	3427
6.14.53	chart.parallel.parallelAxisDefault.max .....	3427
6.14.54	chart.parallel.parallelAxisDefault.scale.....	3428
6.14.55	chart.parallel.parallelAxisDefault.splitNumber .....	3428
6.14.56	chart.parallel.parallelAxisDefault.minInterval.....	3429
6.14.57	chart.parallel.parallelAxisDefault.interval.....	3429
6.14.58	chart.parallel.parallelAxisDefault.logBase .....	3430
6.14.59	chart.parallel.parallelAxisDefault.silent .....	3430
6.14.60	chart.parallel.parallelAxisDefault.triggerEvent.....	3430
6.14.61	chart.parallel.parallelAxisDefault.axisLine.object .....	3431
6.14.62	chart.parallel.parallelAxisDefault.axisLine.show .....	3431
6.14.63	chart.parallel.parallelAxisDefault.axisLine.symbol .....	3432
	Erlaubte Werte .....	3432
6.14.64	chart.parallel.parallelAxisDefault.axisLine.symbolSize.....	3432
6.14.65	chart.parallel.parallelAxisDefault.axisLine.symbolOffset .....	3433
6.14.66	chart.parallel.parallelAxisDefault.axisLine.lineStyle.object.....	3433
6.14.67	chart.parallel.parallelAxisDefault.axisLine.lineStyle.color.....	3433
6.14.68	chart.parallel.parallelAxisDefault.axisLine.lineStyle.width.....	3434
6.14.69	chart.parallel.parallelAxisDefault.axisLine.lineStyle.type.....	3435
	Erlaubte Werte .....	3435
6.14.70	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowBlur .....	3435
6.14.71	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowColor .....	3436
6.14.72	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetX.....	3436
6.14.73	chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetY .....	3436
6.14.74	chart.parallel.parallelAxisDefault.axisLine.lineStyle.opacity.....	3437
6.14.75	chart.parallel.parallelAxisDefault.axisTick.object .....	3437
6.14.76	chart.parallel.parallelAxisDefault.axisTick.show .....	3438
6.14.77	chart.parallel.parallelAxisDefault.axisTick.alignWithLabel.....	3438
6.14.78	chart.parallel.parallelAxisDefault.axisTick.interval .....	3438
6.14.79	chart.parallel.parallelAxisDefault.axisTick.inside.....	3439
6.14.80	chart.parallel.parallelAxisDefault.axisTick.length .....	3439



6.14.81	chart.parallel.parallelAxisDefault.axisTick.lineStyle.object.....	3439
6.14.82	chart.parallel.parallelAxisDefault.axisTick.lineStyle.color.....	3440
6.14.83	chart.parallel.parallelAxisDefault.axisTick.lineStyle.width.....	3440
6.14.84	chart.parallel.parallelAxisDefault.axisTick.lineStyle.type.....	3440
	Erlaubte Werte .....	3441
6.14.85	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowBlur .....	3441
6.14.86	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowColor .....	3441
6.14.87	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetX .....	3442
6.14.88	chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetY .....	3442
6.14.89	chart.parallel.parallelAxisDefault.axisTick.lineStyle.opacity.....	3442
6.14.90	chart.parallel.parallelAxisDefault.axisLabel.object.....	3443
6.14.91	chart.parallel.parallelAxisDefault.axisLabel.show.....	3443
6.14.92	chart.parallel.parallelAxisDefault.axisLabel.interval.....	3443
6.14.93	chart.parallel.parallelAxisDefault.axisLabel.inside.....	3444
6.14.94	chart.parallel.parallelAxisDefault.axisLabel.rotate .....	3444
6.14.95	chart.parallel.parallelAxisDefault.axisLabel.margin.....	3445
6.14.96	chart.parallel.parallelAxisDefault.axisLabel.formatter .....	3445
6.14.97	chart.parallel.parallelAxisDefault.axisLabel.showMinLabel .....	3446
6.14.98	chart.parallel.parallelAxisDefault.axisLabel.showMaxLabel.....	3446
6.14.99	chart.parallel.parallelAxisDefault.axisLabel.color .....	3447
6.14.100	chart.parallel.parallelAxisDefault.axisLabel.fontStyle .....	3447
	Erlaubte Werte .....	3448
6.14.101	chart.parallel.parallelAxisDefault.axisLabel.fontWeight .....	3448
	Erlaubte Werte .....	3448
6.14.102	chart.parallel.parallelAxisDefault.axisLabel.fontFamily.....	3448
6.14.103	chart.parallel.parallelAxisDefault.axisLabel.fontSize.....	3449
6.14.104	chart.parallel.parallelAxisDefault.axisLabel.align .....	3449
	Erlaubte Werte .....	3450
6.14.105	chart.parallel.parallelAxisDefault.axisLabel.verticalAlign.....	3450
	Erlaubte Werte .....	3450
6.14.106	chart.parallel.parallelAxisDefault.axisLabel.lineHeight .....	3451
6.14.107	chart.parallel.parallelAxisDefault.axisLabel.backgroundColor .....	3451
6.14.108	chart.parallel.parallelAxisDefault.axisLabel.borderColor .....	3452
6.14.109	chart.parallel.parallelAxisDefault.axisLabel.borderWidth .....	3452
6.14.110	chart.parallel.parallelAxisDefault.axisLabel.borderRadius.....	3453
6.14.111	chart.parallel.parallelAxisDefault.axisLabel.padding.....	3453

6.14.112	chart.parallel.parallelAxisDefault.axisLabel.shadowColor .....	3453
6.14.113	chart.parallel.parallelAxisDefault.axisLabel.shadowBlur.....	3454
6.14.114	chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetX .....	3454
6.14.115	chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetY.....	3454
6.14.116	chart.parallel.parallelAxisDefault.axisLabel.width .....	3455
6.14.117	chart.parallel.parallelAxisDefault.axisLabel.height .....	3455
6.14.118	chart.parallel.parallelAxisDefault.axisLabel.textBorderColor.....	3456
6.14.119	chart.parallel.parallelAxisDefault.axisLabel.textBorderWidth.....	3456
6.14.120	chart.parallel.parallelAxisDefault.axisLabel.textShadowColor .....	3456
6.14.121	chart.parallel.parallelAxisDefault.axisLabel.textShadowBlur.....	3457
6.14.122	chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetX .....	3457
6.14.123	chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetY.....	3458
6.14.124	chart.parallel.parallelAxisDefault.axisLabel.rich.object.....	3458
6.15	<b>ECharts (5) Parallel Axis .....</b>	<b>3459</b>
6.15.1	chart.parallelAxis.object.....	3459
6.15.2	chart.parallelAxis.id .....	3462
6.15.3	chart.parallelAxis.dim.....	3463
6.15.4	chart.parallelAxis.parallelIndex .....	3463
6.15.5	chart.parallelAxis.realtime .....	3464
6.15.6	chart.parallelAxis.areaSelectStyle.object.....	3465
6.15.7	chart.parallelAxis.areaSelectStyle.width.....	3465
6.15.8	chart.parallelAxis.areaSelectStyle.borderWidth.....	3465
6.15.9	chart.parallelAxis.areaSelectStyle.borderColor.....	3466
6.15.10	chart.parallelAxis.areaSelectStyle.color.....	3466
6.15.11	chart.parallelAxis.areaSelectStyle.opacity.....	3466
6.15.12	chart.parallelAxis.type .....	3467
	Erlaubte Werte .....	3467
6.15.13	chart.parallelAxis.name .....	3467
6.15.14	chart.parallelAxis.nameLocation .....	3468
	Erlaubte Werte .....	3468
6.15.15	chart.parallelAxis.nameTextStyle.object.....	3468
6.15.16	chart.parallelAxis.nameTextStyle.color .....	3469
6.15.17	chart.parallelAxis.nameTextStyle.fontStyle.....	3469
	Erlaubte Werte .....	3469
6.15.18	chart.parallelAxis.nameTextStyle.fontWeight .....	3469
	Erlaubte Werte .....	3470

6.15.19	chart.parallelAxis.nameTextStyle.fontFamily .....	3470
6.15.20	chart.parallelAxis.nameTextStyle.fontSize .....	3470
6.15.21	chart.parallelAxis.nameTextStyle.align .....	3471
	Erlaubte Werte .....	3471
6.15.22	chart.parallelAxis.nameTextStyle.verticalAlign .....	3472
	Erlaubte Werte .....	3472
6.15.23	chart.parallelAxis.nameTextStyle.lineHeight.....	3472
6.15.24	chart.parallelAxis.nameTextStyle.backgroundColor.....	3473
6.15.25	chart.parallelAxis.nameTextStyle.borderColor.....	3473
6.15.26	chart.parallelAxis.nameTextStyle.borderWidth .....	3474
6.15.27	chart.parallelAxis.nameTextStyle.borderRadius .....	3474
6.15.28	chart.parallelAxis.nameTextStyle.padding .....	3475
6.15.29	chart.parallelAxis.nameTextStyle.shadowColor .....	3475
6.15.30	chart.parallelAxis.nameTextStyle.shadowBlur .....	3475
6.15.31	chart.parallelAxis.nameTextStyle.shadowOffsetX.....	3476
6.15.32	chart.parallelAxis.nameTextStyle.shadowOffsetY .....	3476
6.15.33	chart.parallelAxis.nameTextStyle.width .....	3476
6.15.34	chart.parallelAxis.nameTextStyle.height .....	3477
6.15.35	chart.parallelAxis.nameTextStyle.textBorderColor .....	3477
6.15.36	chart.parallelAxis.nameTextStyle.textBorderWidth .....	3478
6.15.37	chart.parallelAxis.nameTextStyle.textShadowColor.....	3478
6.15.38	chart.parallelAxis.nameTextStyle.textShadowBlur .....	3478
6.15.39	chart.parallelAxis.nameTextStyle.textShadowOffsetX.....	3479
6.15.40	chart.parallelAxis.nameTextStyle.textShadowOffsetY .....	3479
6.15.41	chart.parallelAxis.nameTextStyle.rich.object .....	3480
6.15.42	chart.parallelAxis.nameGap .....	3481
6.15.43	chart.parallelAxis.nameRotate.....	3481
6.15.44	chart.parallelAxis.inverse .....	3481
6.15.45	chart.parallelAxis.boundaryGap .....	3482
6.15.46	chart.parallelAxis.min.....	3482
6.15.47	chart.parallelAxis.max .....	3483
6.15.48	chart.parallelAxis.scale.....	3483
6.15.49	chart.parallelAxis.splitNumber .....	3484
6.15.50	chart.parallelAxis.minInterval.....	3484
6.15.51	chart.parallelAxis.interval .....	3484
6.15.52	chart.parallelAxis.logBase .....	3485

6.15.53	chart.parallelAxis.silent .....	3485
6.15.54	chart.parallelAxis.triggerEvent.....	3486
6.15.55	chart.parallelAxis.axisLine.object .....	3486
6.15.56	chart.parallelAxis.axisLine.show .....	3487
6.15.57	chart.parallelAxis.axisLine.symbol .....	3487
	Erlaubte Werte .....	3487
6.15.58	chart.parallelAxis.axisLine.symbolSize.....	3488
6.15.59	chart.parallelAxis.axisLine.symbolOffset .....	3488
6.15.60	chart.parallelAxis.axisLine.lineStyle.object.....	3488
6.15.61	chart.parallelAxis.axisLine.lineStyle.color.....	3489
6.15.62	chart.parallelAxis.axisLine.lineStyle.width .....	3490
6.15.63	chart.parallelAxis.axisLine.lineStyle.type.....	3491
	Erlaubte Werte .....	3491
6.15.64	chart.parallelAxis.axisLine.lineStyle.shadowBlur .....	3491
6.15.65	chart.parallelAxis.axisLine.lineStyle.shadowColor .....	3492
6.15.66	chart.parallelAxis.axisLine.lineStyle.shadowOffsetX.....	3492
6.15.67	chart.parallelAxis.axisLine.lineStyle.shadowOffsetY .....	3492
6.15.68	chart.parallelAxis.axisLine.lineStyle.opacity.....	3493
6.15.69	chart.parallelAxis.axisTick.object .....	3493
6.15.70	chart.parallelAxis.axisTick.show .....	3493
6.15.71	chart.parallelAxis.axisTick.alignWithLabel.....	3494
6.15.72	chart.parallelAxis.axisTick.interval .....	3494
6.15.73	chart.parallelAxis.axisTick.inside.....	3495
6.15.74	chart.parallelAxis.axisTick.length .....	3495
6.15.75	chart.parallelAxis.axisTick.lineStyle.object.....	3495
6.15.76	chart.parallelAxis.axisTick.lineStyle.color.....	3496
6.15.77	chart.parallelAxis.axisTick.lineStyle.width.....	3496
6.15.78	chart.parallelAxis.axisTick.lineStyle.type.....	3496
	Erlaubte Werte .....	3497
6.15.79	chart.parallelAxis.axisTick.lineStyle.shadowBlur .....	3497
6.15.80	chart.parallelAxis.axisTick.lineStyle.shadowColor .....	3497
6.15.81	chart.parallelAxis.axisTick.lineStyle.shadowOffsetX.....	3498
6.15.82	chart.parallelAxis.axisTick.lineStyle.shadowOffsetY .....	3498
6.15.83	chart.parallelAxis.axisTick.lineStyle.opacity.....	3498
6.15.84	chart.parallelAxis.axisLabel.object .....	3499
6.15.85	chart.parallelAxis.axisLabel.show.....	3499

6.15.86	chart.parallelAxis.axisLabel.interval.....	3499
6.15.87	chart.parallelAxis.axisLabel.inside.....	3500
6.15.88	chart.parallelAxis.axisLabel.rotate .....	3500
6.15.89	chart.parallelAxis.axisLabel.margin.....	3500
6.15.90	chart.parallelAxis.axisLabel.formatter .....	3501
6.15.91	chart.parallelAxis.axisLabel.showMinLabel .....	3501
6.15.92	chart.parallelAxis.axisLabel.showMaxLabel.....	3502
6.15.93	chart.parallelAxis.axisLabel.color .....	3502
6.15.94	chart.parallelAxis.axisLabel.fontStyle .....	3503
	Erlaubte Werte .....	3503
6.15.95	chart.parallelAxis.axisLabel.fontWeight .....	3503
	Erlaubte Werte .....	3504
6.15.96	chart.parallelAxis.axisLabel.fontFamily .....	3504
6.15.97	chart.parallelAxis.axisLabel.fontSize .....	3504
6.15.98	chart.parallelAxis.axisLabel.align .....	3505
	Erlaubte Werte .....	3505
6.15.99	chart.parallelAxis.axisLabel.verticalAlign.....	3505
	Erlaubte Werte .....	3506
6.15.100	chart.parallelAxis.axisLabel.lineHeight .....	3506
6.15.101	chart.parallelAxis.axisLabel.backgroundColor .....	3507
6.15.102	chart.parallelAxis.axisLabel.borderColor .....	3507
6.15.103	chart.parallelAxis.axisLabel.borderWidth .....	3508
6.15.104	chart.parallelAxis.axisLabel.borderRadius.....	3508
6.15.105	chart.parallelAxis.axisLabel.padding.....	3508
6.15.106	chart.parallelAxis.axisLabel.shadowColor .....	3509
6.15.107	chart.parallelAxis.axisLabel.shadowBlur .....	3509
6.15.108	chart.parallelAxis.axisLabel.shadowOffsetX .....	3510
6.15.109	chart.parallelAxis.axisLabel.shadowOffsetY .....	3510
6.15.110	chart.parallelAxis.axisLabel.width .....	3510
6.15.111	chart.parallelAxis.axisLabel.height.....	3511
6.15.112	chart.parallelAxis.axisLabel.textBorderColor.....	3511
6.15.113	chart.parallelAxis.axisLabel.textBorderWidth.....	3512
6.15.114	chart.parallelAxis.axisLabel.textShadowColor .....	3512
6.15.115	chart.parallelAxis.axisLabel.textShadowBlur.....	3512
6.15.116	chart.parallelAxis.axisLabel.textShadowOffsetX .....	3513
6.15.117	chart.parallelAxis.axisLabel.textShadowOffsetY.....	3513

6.15.118	chart.parallelAxis.axisLabel.rich.object.....	3513
<b>6.16</b>	<b>ECharts (5) Polar.....</b>	<b>3514</b>
6.16.1	chart.polar.object .....	3514
6.16.2	chart.polar.id.....	3515
6.16.3	chart.polar.zlevel .....	3515
6.16.4	chart.polar.z .....	3516
6.16.5	chart.polar.center .....	3516
6.16.6	chart.polar.radius .....	3517
6.16.7	chart.polar.tooltip.object.....	3517
6.16.8	chart.polar.tooltip.show.....	3518
6.16.9	chart.polar.tooltip.trigger .....	3518
	Erlaubte Werte .....	3518
6.16.10	chart.polar.tooltip.axisPointer.object .....	3519
6.16.11	chart.polar.tooltip.axisPointer.type .....	3519
	Erlaubte Werte .....	3520
6.16.12	chart.polar.tooltip.axisPointer.axis .....	3520
	Erlaubte Werte .....	3520
6.16.13	chart.polar.tooltip.axisPointer.snap.....	3520
6.16.14	chart.polar.tooltip.axisPointer.z .....	3521
6.16.15	chart.polar.tooltip.axisPointer.label.object.....	3521
6.16.16	chart.polar.tooltip.axisPointer.label.show .....	3521
6.16.17	chart.polar.tooltip.axisPointer.label.precision .....	3522
6.16.18	chart.polar.tooltip.axisPointer.label.formatter .....	3522
6.16.19	chart.polar.tooltip.axisPointer.label.margin .....	3524
6.16.20	chart.polar.tooltip.axisPointer.label.color .....	3524
6.16.21	chart.polar.tooltip.axisPointer.label.fontStyle .....	3524
6.16.22	chart.polar.tooltip.axisPointer.label.fontWeight.....	3525
	Erlaubte Werte .....	3525
6.16.23	chart.polar.tooltip.axisPointer.label.fontFamily .....	3525
6.16.24	chart.polar.tooltip.axisPointer.label.fontSize.....	3526
6.16.25	chart.polar.tooltip.axisPointer.label.lineHeight .....	3526
6.16.26	chart.polar.tooltip.axisPointer.label.width.....	3527
6.16.27	chart.polar.tooltip.axisPointer.label.height.....	3527
6.16.28	chart.polar.tooltip.axisPointer.label.textBorderColor .....	3527
6.16.29	chart.polar.tooltip.axisPointer.label.textBorderWidth .....	3528
6.16.30	chart.polar.tooltip.axisPointer.label.textShadowColor .....	3528

6.16.31	chart.polar.tooltip.axisPointer.label.textShadowBlur .....	3529
6.16.32	chart.polar.tooltip.axisPointer.label.textShadowOffsetX .....	3529
6.16.33	chart.polar.tooltip.axisPointer.label.textShadowOffsetY .....	3529
6.16.34	chart.polar.tooltip.axisPointer.label.padding .....	3530
6.16.35	chart.polar.tooltip.axisPointer.label.backgroundColor .....	3530
6.16.36	chart.polar.tooltip.axisPointer.label.borderColor .....	3531
6.16.37	chart.polar.tooltip.axisPointer.label.borderWidth .....	3531
6.16.38	chart.polar.tooltip.axisPointer.label.shadowBlur .....	3531
6.16.39	chart.polar.tooltip.axisPointer.label.shadowColor .....	3532
6.16.40	chart.polar.tooltip.axisPointer.label.shadowOffsetX .....	3532
6.16.41	chart.polar.tooltip.axisPointer.label.shadowOffsetY .....	3533
6.16.42	chart.polar.tooltip.axisPointer.lineStyle.object.....	3533
6.16.43	chart.polar.tooltip.axisPointer.lineStyle.color .....	3533
6.16.44	chart.polar.tooltip.axisPointer.lineStyle.width .....	3535
6.16.45	chart.polar.tooltip.axisPointer.lineStyle.type.....	3535
	Erlaubte Werte .....	3535
6.16.46	chart.polar.tooltip.axisPointer.lineStyle.shadowBlur .....	3535
6.16.47	chart.polar.tooltip.axisPointer.lineStyle.shadowColor .....	3536
6.16.48	chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetX.....	3536
6.16.49	chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetY .....	3537
6.16.50	chart.polar.tooltip.axisPointer.lineStyle.opacity.....	3537
6.16.51	chart.polar.tooltip.axisPointer.shadowStyle.object.....	3537
6.16.52	chart.polar.tooltip.axisPointer.shadowStyle.color.....	3538
6.16.53	chart.polar.tooltip.axisPointer.shadowStyle.shadowBlur .....	3539
6.16.54	chart.polar.tooltip.axisPointer.shadowStyle.shadowColor .....	3540
6.16.55	chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetX.....	3540
6.16.56	chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	3541
6.16.57	chart.polar.tooltip.axisPointer.shadowStyle.opacity.....	3541
6.16.58	chart.polar.tooltip.axisPointer.crossStyle.object .....	3541
6.16.59	chart.polar.tooltip.axisPointer.crossStyle.color .....	3542
6.16.60	chart.polar.tooltip.axisPointer.crossStyle.width .....	3543
6.16.61	chart.polar.tooltip.axisPointer.crossStyle.type .....	3544
	Erlaubte Werte .....	3544
6.16.62	chart.polar.tooltip.axisPointer.crossStyle.shadowBlur .....	3544
6.16.63	chart.polar.tooltip.axisPointer.crossStyle.shadowColor .....	3545
6.16.64	chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetX .....	3545

6.16.65	chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetY .....	3545
6.16.66	chart.polar.tooltip.axisPointer.crossStyle.opacity .....	3546
6.16.67	chart.polar.tooltip.position .....	3546
6.16.68	chart.polar.tooltip.formatter .....	3548
6.16.69	chart.polar.tooltip.backgroundColor .....	3550
6.16.70	chart.polar.tooltip.borderColor .....	3550
6.16.71	chart.polar.tooltip.borderWidth .....	3551
6.16.72	chart.polar.tooltip.padding .....	3551
6.16.73	chart.polar.tooltip.textStyle.object .....	3552
6.16.74	chart.polar.tooltip.textStyle.color .....	3552
6.16.75	chart.polar.tooltip.textStyle.fontStyle .....	3552
	Erlaubte Werte .....	3553
6.16.76	chart.polar.tooltip.textStyle.fontWeight .....	3553
	Erlaubte Werte .....	3553
6.16.77	chart.polar.tooltip.textStyle.fontFamily .....	3553
6.16.78	chart.polar.tooltip.textStyle.fontSize .....	3554
6.16.79	chart.polar.tooltip.textStyle.lineHeight .....	3554
6.16.80	chart.polar.tooltip.textStyle.width .....	3555
6.16.81	chart.polar.tooltip.textStyle.height .....	3555
6.16.82	chart.polar.tooltip.textStyle.textBorderColor .....	3556
6.16.83	chart.polar.tooltip.textStyle.textBorderWidth .....	3556
6.16.84	chart.polar.tooltip.textStyle.textShadowColor .....	3556
6.16.85	chart.polar.tooltip.textStyle.textShadowBlur .....	3557
6.16.86	chart.polar.tooltip.textStyle.textShadowOffsetX .....	3557
6.16.87	chart.polar.tooltip.textStyle.textShadowOffsetY .....	3557
6.16.88	chart.polar.tooltip.extraCssText .....	3558
6.16.89	chart.polar.tooltip.axisPointer.label.ellipsis .....	3558
6.16.90	chart.polar.tooltip.axisPointer.label.overflow .....	3559
	Erlaubte Werte .....	3559
6.16.91	chart.polar.tooltip.textStyle.overflow .....	3559
	Erlaubte Werte .....	3559
6.16.92	chart.polar.tooltip.textStyle.ellipsis .....	3560
6.16.93	chart.polar.tooltip.axisPointer.animation .....	3560
6.16.94	chart.polar.tooltip.axisPointer.animationThreshold .....	3560
6.16.95	chart.polar.tooltip.axisPointer.animationDuration .....	3561
6.16.96	chart.polar.tooltip.axisPointer.animationEasing .....	3561



	Erlaubte Werte .....	3562
6.16.97	chart.polar.tooltip.axisPointer.animationDelay .....	3562
6.16.98	chart.polar.tooltip.axisPointer.animationDurationUpdate .....	3562
6.16.99	chart.polar.tooltip.axisPointer.animationEasingUpdate .....	3563
	Erlaubte Werte .....	3563
6.16.100	chart.polar.tooltip.axisPointer.animationDelayUpdate.....	3563
<b>6.17</b>	<b>ECharts (5) Radar .....</b>	<b>3564</b>
6.17.1	chart.radar.object .....	3564
6.17.2	chart.radar.id .....	3564
6.17.3	chart.radar.zlevel .....	3565
6.17.4	chart.radar.z .....	3565
6.17.5	chart.radar.center .....	3566
6.17.6	chart.radar.radius .....	3566
6.17.7	chart.radar.startAngle .....	3567
6.17.8	chart.radar.name.object.....	3567
6.17.9	chart.radar.name.show .....	3567
6.17.10	chart.radar.name.formatter .....	3568
6.17.11	chart.radar.name.color.....	3568
6.17.12	chart.radar.name.fontStyle .....	3569
	Erlaubte Werte .....	3569
6.17.13	chart.radar.name.fontWeight .....	3569
	Erlaubte Werte .....	3570
6.17.14	chart.radar.name.fontFamily .....	3570
6.17.15	chart.radar.name.fontSize .....	3570
6.17.16	chart.radar.name.lineHeight.....	3570
6.17.17	chart.radar.name.backgroundColor.....	3571
6.17.18	chart.radar.name.borderColor.....	3572
6.17.19	chart.radar.name.borderWidth.....	3572
6.17.20	chart.radar.name.borderRadius .....	3572
6.17.21	chart.radar.name.padding .....	3573
6.17.22	chart.radar.name.shadowColor .....	3573
6.17.23	chart.radar.name.shadowBlur .....	3574
6.17.24	chart.radar.name.shadowOffsetX.....	3574
6.17.25	chart.radar.name.shadowOffsetY .....	3574
6.17.26	chart.radar.name.width .....	3575
	Erlaubte Werte .....	3575

6.17.27	chart.radar.name.height .....	3575
6.17.28	chart.radar.name.textBorderColor .....	3576
6.17.29	chart.radar.name.textBorderWidth .....	3576
6.17.30	chart.radar.name.textShadowColor .....	3576
6.17.31	chart.radar.name.textShadowBlur .....	3577
6.17.32	chart.radar.name.textShadowOffsetX .....	3577
6.17.33	chart.radar.name.textShadowOffsetY .....	3577
6.17.34	chart.radar.name.rich.object .....	3578
6.17.35	chart.radar.nameGap .....	3579
6.17.36	chart.radar.splitNumber .....	3580
6.17.37	chart.radar.shape .....	3580
	Erlaubte Werte .....	3580
6.17.38	chart.radar.scale .....	3580
6.17.39	chart.radar.silent .....	3581
6.17.40	chart.radar.triggerEvent .....	3581
6.17.41	chart.radar.axisLine.object .....	3582
6.17.42	chart.radar.axisLine.show .....	3582
6.17.43	chart.radar.axisLine.symbol .....	3583
	Erlaubte Werte .....	3583
6.17.44	chart.radar.axisLine.symbolSize .....	3583
6.17.45	chart.radar.axisLine.symbolOffset .....	3583
6.17.46	chart.radar.axisLine.lineStyle.object .....	3584
6.17.47	chart.radar.axisLine.lineStyle.color .....	3584
6.17.48	chart.radar.axisLine.lineStyle.width .....	3585
6.17.49	chart.radar.axisLine.lineStyle.type .....	3586
	Erlaubte Werte .....	3586
6.17.50	chart.radar.axisLine.lineStyle.shadowBlur .....	3586
6.17.51	chart.radar.axisLine.lineStyle.shadowColor .....	3587
6.17.52	chart.radar.axisLine.lineStyle.shadowOffsetX .....	3587
6.17.53	chart.radar.axisLine.lineStyle.shadowOffsetY .....	3587
6.17.54	chart.radar.axisLine.lineStyle.opacity .....	3588
6.17.55	chart.radar.splitLine.object .....	3588
6.17.56	chart.radar.splitLine.show .....	3588
6.17.57	chart.radar.splitLine.lineStyle.object .....	3589
6.17.58	chart.radar.splitLine.lineStyle.color .....	3589
6.17.59	chart.radar.splitLine.lineStyle.width .....	3590

6.17.60	chart.radar.splitLine.lineStyle.type .....	3590
	Erlaubte Werte .....	3590
6.17.61	chart.radar.splitLine.lineStyle.shadowBlur .....	3591
6.17.62	chart.radar.splitLine.lineStyle.shadowColor .....	3591
6.17.63	chart.radar.splitLine.lineStyle.shadowOffsetX .....	3591
6.17.64	chart.radar.splitLine.lineStyle.shadowOffsetY.....	3592
6.17.65	chart.radar.splitLine.lineStyle.opacity .....	3592
6.17.66	chart.radar.splitArea.object .....	3592
6.17.67	chart.radar.splitArea.show .....	3593
6.17.68	chart.radar.splitArea.areaStyle.object .....	3593
6.17.69	chart.radar.splitArea.areaStyle.color .....	3594
6.17.70	chart.radar.splitArea.areaStyle.shadowBlur .....	3594
6.17.71	chart.radar.splitArea.areaStyle.shadowColor .....	3594
6.17.72	chart.radar.splitArea.areaStyle.shadowOffsetX.....	3595
6.17.73	chart.radar.splitArea.areaStyle.shadowOffsetY.....	3595
6.17.74	chart.radar.splitArea.areaStyle.opacity .....	3595
6.17.75	chart.radar.indicator.object .....	3596
6.17.76	chart.radar.color .....	3596
6.17.77	chart.radar.axisName.object.....	3597
6.17.78	chart.radar.axisName.show .....	3597
6.17.79	chart.radar.axisName.formatter .....	3597
6.17.80	chart.radar.axisName.color.....	3598
6.17.81	chart.radar.axisName.fontStyle .....	3598
	Erlaubte Werte .....	3599
6.17.82	chart.radar.axisName.fontWeight.....	3599
	Erlaubte Werte .....	3599
6.17.83	chart.radar.axisName.fontFamily .....	3599
6.17.84	chart.radar.axisName.fontSize .....	3600
6.17.85	chart.radar.axisName.lineHeight.....	3600
6.17.86	chart.radar.axisName.backgroundColor .....	3601
6.17.87	chart.radar.axisName.borderColor .....	3601
6.17.88	chart.radar.axisName.borderWidth .....	3602
6.17.89	chart.radar.axisName.borderType.....	3602
	Erlaubte Werte .....	3602
6.17.90	chart.radar.axisName.borderDashOffset .....	3603
6.17.91	chart.radar.axisName.borderRadius .....	3603

6.17.92	chart.radar.axisName.padding .....	3603
6.17.93	chart.radar.axisName.shadowColor .....	3604
6.17.94	chart.radar.axisName.shadowBlur .....	3604
6.17.95	chart.radar.axisName.shadowOffsetX .....	3604
6.17.96	chart.radar.axisName.shadowOffsetY .....	3605
6.17.97	chart.radar.axisName.width.....	3605
	Erlaubte Werte .....	3606
6.17.98	chart.radar.axisName.height .....	3606
6.17.99	chart.radar.axisName.textBorderColor .....	3606
6.17.100	chart.radar.axisName.textBorderWidth .....	3606
6.17.101	chart.radar.axisName.textBorderStyle .....	3607
	Erlaubte Werte .....	3607
6.17.102	chart.radar.axisName.textBorderDashOffset.....	3607
6.17.103	chart.radar.axisName.textShadowColor .....	3608
6.17.104	chart.radar.axisName.textShadowBlur .....	3608
6.17.105	chart.radar.axisName.textShadowOffsetX.....	3609
6.17.106	chart.radar.axisName.textShadowOffsetY .....	3609
6.17.107	chart.radar.axisName.overflow.....	3609
	Erlaubte Werte .....	3610
6.17.108	chart.radar.axisName.ellipsis.....	3610
6.17.109	chart.radar.axisName.rich.object .....	3610
6.17.110	chart.radar.axisLine.lineStyle.dashOffset .....	3611
6.17.111	chart.radar.axisLine.lineStyle.cap .....	3612
	Erlaubte Werte .....	3612
6.17.112	chart.radar.axisLine.lineStyle.join .....	3612
	Erlaubte Werte .....	3613
6.17.113	chart.radar.axisLine.lineStyle.miterLimit.....	3613
6.17.114	chart.radar.axisTick.object.....	3613
6.17.115	chart.radar.axisTick.show .....	3614
6.17.116	chart.radar.axisTick.length .....	3614
6.17.117	chart.radar.axisTick.lineStyle.object.....	3614
6.17.118	chart.radar.axisTick.lineStyle.color .....	3615
6.17.119	chart.radar.axisTick.lineStyle.width .....	3615
6.17.120	chart.radar.axisTick.lineStyle.type .....	3615
	Erlaubte Werte .....	3616
6.17.121	chart.radar.axisTick.lineStyle.dashOffset .....	3616

6.17.122	chart.radar.axisTick.lineStyle.shadowBlur .....	3616
6.17.123	chart.radar.axisTick.lineStyle.shadowColor .....	3617
6.17.124	chart.radar.axisTick.lineStyle.shadowOffsetX .....	3617
6.17.125	chart.radar.axisTick.lineStyle.shadowOffsetY .....	3617
6.17.126	chart.radar.axisTick.lineStyle.opacity .....	3618
6.17.127	chart.radar.axisLabel.object .....	3618
6.17.128	chart.radar.axisLabel.show .....	3618
6.17.129	chart.radar.axisLabel.rotate.....	3619
6.17.130	chart.radar.axisLabel.margin .....	3619
6.17.131	chart.radar.axisLabel.formatter.....	3619
6.17.132	chart.radar.axisLabel.showMinLabel.....	3620
6.17.133	chart.radar.axisLabel.showMaxLabel .....	3620
6.17.134	chart.radar.axisLabel.color .....	3621
6.17.135	chart.radar.axisLabel.fontStyle.....	3621
	Erlaubte Werte .....	3621
6.17.136	chart.radar.axisLabel.fontWeight .....	3622
	Erlaubte Werte .....	3622
6.17.137	chart.radar.axisLabel.fontFamily.....	3622
6.17.138	chart.radar.axisLabel.fontSize .....	3623
6.17.139	chart.radar.axisLabel.align.....	3623
	Erlaubte Werte .....	3624
6.17.140	chart.radar.axisLabel.verticalAlign .....	3624
	Erlaubte Werte .....	3624
6.17.141	chart.radar.axisLabel.lineHeight.....	3624
6.17.142	chart.radar.axisLabel.backgroundColor.....	3625
6.17.143	chart.radar.axisLabel.borderColor .....	3626
6.17.144	chart.radar.axisLabel.borderWidth.....	3626
6.17.145	chart.radar.axisLabel.borderRadius .....	3626
6.17.146	chart.radar.axisLabel.borderType .....	3627
	Erlaubte Werte .....	3627
6.17.147	chart.radar.axisLabel.borderDashOffset .....	3627
6.17.148	chart.radar.axisLabel.padding.....	3627
6.17.149	chart.radar.axisLabel.shadowColor.....	3628
6.17.150	chart.radar.axisLabel.shadowBlur.....	3628
6.17.151	chart.radar.axisLabel.shadowOffsetX.....	3629
6.17.152	chart.radar.axisLabel.shadowOffsetY.....	3629

6.17.153	chart.radar.axisLabel.width .....	3629
6.17.154	chart.radar.axisLabel.height .....	3630
6.17.155	chart.radar.axisLabel.textBorderColor .....	3630
6.17.156	chart.radar.axisLabel.textBorderType .....	3630
	Erlaubte Werte .....	3631
6.17.157	chart.radar.axisLabel.textBorderDashOffset .....	3631
6.17.158	chart.radar.axisLabel.textBorderWidth .....	3631
6.17.159	chart.radar.axisLabel.textShadowColor .....	3632
6.17.160	chart.radar.axisLabel.textShadowBlur .....	3632
6.17.161	chart.radar.axisLabel.textShadowOffsetX .....	3632
6.17.162	chart.radar.axisLabel.textShadowOffsetY .....	3633
6.17.163	chart.radar.axisLabel.overflow .....	3633
	Erlaubte Werte .....	3634
6.17.164	chart.radar.axisLabel.ellipsis .....	3634
6.17.165	chart.radar.axisLabel.rich.object .....	3634
6.17.166	chart.radar.splitLine.lineStyle.dashOffset .....	3635
6.17.167	chart.radar.splitLine.lineStyle.cap .....	3636
	Erlaubte Werte .....	3636
6.17.168	chart.radar.splitLine.lineStyle.join .....	3636
	Erlaubte Werte .....	3637
6.17.169	chart.radar.splitLine.lineStyle.miterLimit .....	3637
6.17.170	chart.radar.indicator.color .....	3637
6.17.171	chart.radar.indicator.name .....	3638
6.17.172	chart.radar.indicator.max .....	3638
6.17.173	chart.radar.indicator.min .....	3638
6.18	<b>ECharts (5) Radius Axis.....</b>	<b>3639</b>
6.18.1	chart.radiusAxis.object .....	3639
6.18.2	chart.radiusAxis.id .....	3639
6.18.3	chart.radiusAxis.polarIndex .....	3639
6.18.4	chart.radiusAxis.type .....	3640
	Erlaubte Werte .....	3640
6.18.5	chart.radiusAxis.name .....	3640
6.18.6	chart.radiusAxis.nameLocation .....	3641
	Erlaubte Werte .....	3641
6.18.7	chart.radiusAxis.nameTextStyle.object .....	3641
6.18.8	chart.radiusAxis.nameTextStyle.color .....	3641

6.18.9	chart.radiusAxis.nameTextStyle.fontStyle .....	3642
	Erlaubte Werte .....	3642
6.18.10	chart.radiusAxis.nameTextStyle.fontWeight .....	3642
	Erlaubte Werte .....	3643
6.18.11	chart.radiusAxis.nameTextStyle.fontFamily .....	3643
6.18.12	chart.radiusAxis.nameTextStyle.fontSize .....	3643
6.18.13	chart.radiusAxis.nameTextStyle.align .....	3644
	Erlaubte Werte .....	3644
6.18.14	chart.radiusAxis.nameTextStyle.verticalAlign .....	3644
	Erlaubte Werte .....	3645
6.18.15	chart.radiusAxis.nameTextStyle.lineHeight .....	3645
6.18.16	chart.radiusAxis.nameTextStyle.backgroundColor .....	3645
6.18.17	chart.radiusAxis.nameTextStyle.borderColor .....	3646
6.18.18	chart.radiusAxis.nameTextStyle.borderWidth .....	3646
6.18.19	chart.radiusAxis.nameTextStyle.borderRadius .....	3647
6.18.20	chart.radiusAxis.nameTextStyle.padding .....	3647
6.18.21	chart.radiusAxis.nameTextStyle.shadowColor .....	3648
6.18.22	chart.radiusAxis.nameTextStyle.shadowBlur .....	3648
6.18.23	chart.radiusAxis.nameTextStyle.shadowOffsetX .....	3648
6.18.24	chart.radiusAxis.nameTextStyle.shadowOffsetY .....	3649
6.18.25	chart.radiusAxis.nameTextStyle.width .....	3649
6.18.26	chart.radiusAxis.nameTextStyle.height .....	3650
6.18.27	chart.radiusAxis.nameTextStyle.textBorderColor .....	3650
6.18.28	chart.radiusAxis.nameTextStyle.textBorderWidth .....	3650
6.18.29	chart.radiusAxis.nameTextStyle.textShadowColor .....	3651
6.18.30	chart.radiusAxis.nameTextStyle.textShadowBlur .....	3651
6.18.31	chart.radiusAxis.nameTextStyle.textShadowOffsetX .....	3651
6.18.32	chart.radiusAxis.nameTextStyle.textShadowOffsetY .....	3652
6.18.33	chart.radiusAxis.nameTextStyle.rich.object .....	3652
6.18.34	chart.radiusAxis.nameGap .....	3653
6.18.35	chart.radiusAxis.nameRotate .....	3654
6.18.36	chart.radiusAxis.inverse .....	3654
6.18.37	chart.radiusAxis.boundaryGap .....	3654
6.18.38	chart.radiusAxis.min .....	3655
6.18.39	chart.radiusAxis.max .....	3655
6.18.40	chart.radiusAxis.scale .....	3656

6.18.41	chart.radiusAxis.splitNumber.....	3656
6.18.42	chart.radiusAxis.minInterval .....	3657
6.18.43	chart.radiusAxis.interval.....	3657
6.18.44	chart.radiusAxis.logBase .....	3658
6.18.45	chart.radiusAxis.silent .....	3658
6.18.46	chart.radiusAxis.triggerEvent.....	3658
6.18.47	chart.radiusAxis.axisLine.object .....	3659
6.18.48	chart.radiusAxis.axisLine.show .....	3659
6.18.49	chart.radiusAxis.axisLine.symbol.....	3660
	Erlaubte Werte .....	3660
6.18.50	chart.radiusAxis.axisLine.symbolSize.....	3660
6.18.51	chart.radiusAxis.axisLine.symbolOffset.....	3661
6.18.52	chart.radiusAxis.axisLine.lineStyle.object.....	3661
6.18.53	chart.radiusAxis.axisLine.lineStyle.color.....	3661
6.18.54	chart.radiusAxis.axisLine.lineStyle.width.....	3662
6.18.55	chart.radiusAxis.axisLine.lineStyle.type.....	3663
	Erlaubte Werte .....	3663
6.18.56	chart.radiusAxis.axisLine.lineStyle.shadowBlur .....	3663
6.18.57	chart.radiusAxis.axisLine.lineStyle.shadowColor .....	3664
6.18.58	chart.radiusAxis.axisLine.lineStyle.shadowOffsetX .....	3664
6.18.59	chart.radiusAxis.axisLine.lineStyle.shadowOffsetY .....	3664
6.18.60	chart.radiusAxis.axisLine.lineStyle.opacity.....	3665
6.18.61	chart.radiusAxis.axisTick.object.....	3665
6.18.62	chart.radiusAxis.axisTick.show .....	3666
6.18.63	chart.radiusAxis.axisTick.alignWithLabel .....	3666
6.18.64	chart.radiusAxis.axisTick.interval .....	3666
6.18.65	chart.radiusAxis.axisTick.inside .....	3667
6.18.66	chart.radiusAxis.axisTick.length .....	3667
6.18.67	chart.radiusAxis.axisTick.lineStyle.object.....	3667
6.18.68	chart.radiusAxis.axisTick.lineStyle.color.....	3668
6.18.69	chart.radiusAxis.axisTick.lineStyle.width.....	3668
6.18.70	chart.radiusAxis.axisTick.lineStyle.type.....	3668
	Erlaubte Werte .....	3669
6.18.71	chart.radiusAxis.axisTick.lineStyle.shadowBlur .....	3669
6.18.72	chart.radiusAxis.axisTick.lineStyle.shadowColor .....	3669
6.18.73	chart.radiusAxis.axisTick.lineStyle.shadowOffsetX .....	3670



6.18.74	chart.radiusAxis.axisTick.lineStyle.shadowOffsetY .....	3670
6.18.75	chart.radiusAxis.axisTick.lineStyle.opacity .....	3670
6.18.76	chart.radiusAxis.axisLabel.object .....	3671
6.18.77	chart.radiusAxis.axisLabel.show .....	3671
6.18.78	chart.radiusAxis.axisLabel.interval .....	3671
6.18.79	chart.radiusAxis.axisLabel.inside .....	3672
6.18.80	chart.radiusAxis.axisLabel.rotate .....	3672
6.18.81	chart.radiusAxis.axisLabel.margin .....	3673
6.18.82	chart.radiusAxis.axisLabel.formatter .....	3673
6.18.83	chart.radiusAxis.axisLabel.showMinLabel .....	3674
6.18.84	chart.radiusAxis.axisLabel.showMaxLabel .....	3674
6.18.85	chart.radiusAxis.axisLabel.color .....	3675
6.18.86	chart.radiusAxis.axisLabel.fontStyle .....	3675
	Erlaubte Werte .....	3676
6.18.87	chart.radiusAxis.axisLabel.fontWeight .....	3676
	Erlaubte Werte .....	3676
6.18.88	chart.radiusAxis.axisLabel.fontFamily .....	3676
6.18.89	chart.radiusAxis.axisLabel.fontSize .....	3677
6.18.90	chart.radiusAxis.axisLabel.align .....	3677
	Erlaubte Werte .....	3678
6.18.91	chart.radiusAxis.axisLabel.verticalAlign .....	3678
	Erlaubte Werte .....	3678
6.18.92	chart.radiusAxis.axisLabel.lineHeight .....	3679
6.18.93	chart.radiusAxis.axisLabel.backgroundColor .....	3679
6.18.94	chart.radiusAxis.axisLabel.borderColor .....	3680
6.18.95	chart.radiusAxis.axisLabel.borderWidth .....	3680
6.18.96	chart.radiusAxis.axisLabel.borderRadius .....	3681
6.18.97	chart.radiusAxis.axisLabel.padding .....	3681
6.18.98	chart.radiusAxis.axisLabel.shadowColor .....	3681
6.18.99	chart.radiusAxis.axisLabel.shadowBlur .....	3682
6.18.100	chart.radiusAxis.axisLabel.shadowOffsetX .....	3682
6.18.101	chart.radiusAxis.axisLabel.shadowOffsetY .....	3682
6.18.102	chart.radiusAxis.axisLabel.width .....	3683
6.18.103	chart.radiusAxis.axisLabel.height .....	3683
6.18.104	chart.radiusAxis.axisLabel.textBorderColor .....	3684
6.18.105	chart.radiusAxis.axisLabel.textBorderWidth .....	3684

6.18.106	chart.radiusAxis.axisLabel.textShadowColor.....	3684
6.18.107	chart.radiusAxis.axisLabel.textShadowBlur.....	3685
6.18.108	chart.radiusAxis.axisLabel.textShadowOffsetX.....	3685
6.18.109	chart.radiusAxis.axisLabel.textShadowOffsetY.....	3686
6.18.110	chart.radiusAxis.axisLabel.rich.object.....	3686
6.18.111	chart.radiusAxis.splitLine.object.....	3687
6.18.112	chart.radiusAxis.splitLine.show.....	3688
6.18.113	chart.radiusAxis.splitLine.interval.....	3688
6.18.114	chart.radiusAxis.splitLine.lineStyle.object.....	3688
6.18.115	chart.radiusAxis.splitLine.lineStyle.color.....	3689
6.18.116	chart.radiusAxis.splitLine.lineStyle.width.....	3689
6.18.117	chart.radiusAxis.splitLine.lineStyle.type.....	3690
	Erlaubte Werte.....	3690
6.18.118	chart.radiusAxis.splitLine.lineStyle.shadowBlur.....	3690
6.18.119	chart.radiusAxis.splitLine.lineStyle.shadowColor.....	3691
6.18.120	chart.radiusAxis.splitLine.lineStyle.shadowOffsetX.....	3691
6.18.121	chart.radiusAxis.splitLine.lineStyle.shadowOffsetY.....	3691
6.18.122	chart.radiusAxis.splitLine.lineStyle.opacity.....	3692
6.18.123	chart.radiusAxis.splitArea.object.....	3692
6.18.124	chart.radiusAxis.splitArea.interval.....	3692
6.18.125	chart.radiusAxis.splitArea.show.....	3693
6.18.126	chart.radiusAxis.splitArea.areaStyle.object.....	3693
6.18.127	chart.radiusAxis.splitArea.areaStyle.color.....	3693
6.18.128	chart.radiusAxis.splitArea.areaStyle.shadowBlur.....	3694
6.18.129	chart.radiusAxis.splitArea.areaStyle.shadowColor.....	3694
6.18.130	chart.radiusAxis.splitArea.areaStyle.shadowOffsetX.....	3695
6.18.131	chart.radiusAxis.splitArea.areaStyle.shadowOffsetY.....	3695
6.18.132	chart.radiusAxis.splitArea.areaStyle.opacity.....	3695
6.18.133	chart.radiusAxis.axisPointer.object.....	3696
6.18.134	chart.radiusAxis.axisPointer.show.....	3696
6.18.135	chart.radiusAxis.axisPointer.type.....	3696
	Erlaubte Werte.....	3697
6.18.136	chart.radiusAxis.axisPointer.snap.....	3697
6.18.137	chart.radiusAxis.axisPointer.z.....	3697
6.18.138	chart.radiusAxis.axisPointer.label.object.....	3698
6.18.139	chart.radiusAxis.axisPointer.label.show.....	3698

6.18.140	chart.radiusAxis.axisPointer.label.precision .....	3698
6.18.141	chart.radiusAxis.axisPointer.label.formatter .....	3699
6.18.142	chart.radiusAxis.axisPointer.label.margin .....	3700
6.18.143	chart.radiusAxis.axisPointer.label.color .....	3701
6.18.144	chart.radiusAxis.axisPointer.label.fontStyle .....	3701
	Erlaubte Werte .....	3701
6.18.145	chart.radiusAxis.axisPointer.label.fontWeight .....	3702
	Erlaubte Werte .....	3702
6.18.146	chart.radiusAxis.axisPointer.label.fontFamily .....	3702
6.18.147	chart.radiusAxis.axisPointer.label.fontSize .....	3703
6.18.148	chart.radiusAxis.axisPointer.label.lineHeight .....	3703
6.18.149	chart.radiusAxis.axisPointer.label.width .....	3704
6.18.150	chart.radiusAxis.axisPointer.label.height .....	3704
6.18.151	chart.radiusAxis.axisPointer.label.textBorderColor .....	3704
6.18.152	chart.radiusAxis.axisPointer.label.textBorderWidth .....	3705
6.18.153	chart.radiusAxis.axisPointer.label.textShadowColor .....	3705
6.18.154	chart.radiusAxis.axisPointer.label.textShadowBlur .....	3706
6.18.155	chart.radiusAxis.axisPointer.label.textShadowOffsetX .....	3706
6.18.156	chart.radiusAxis.axisPointer.label.textShadowOffsetY .....	3706
6.18.157	chart.radiusAxis.axisPointer.label.padding .....	3707
6.18.158	chart.radiusAxis.axisPointer.label.backgroundColor .....	3707
6.18.159	chart.radiusAxis.axisPointer.label.borderColor .....	3708
6.18.160	chart.radiusAxis.axisPointer.label.borderWidth .....	3708
6.18.161	chart.radiusAxis.axisPointer.label.shadowBlur .....	3708
6.18.162	chart.radiusAxis.axisPointer.label.shadowColor .....	3709
6.18.163	chart.radiusAxis.axisPointer.label.shadowOffsetX .....	3709
6.18.164	chart.radiusAxis.axisPointer.label.shadowOffsetY .....	3710
6.18.165	chart.radiusAxis.axisPointer.lineStyle.object .....	3710
6.18.166	chart.radiusAxis.axisPointer.lineStyle.color .....	3710
6.18.167	chart.radiusAxis.axisPointer.lineStyle.width .....	3712
6.18.168	chart.radiusAxis.axisPointer.lineStyle.type .....	3712
	Erlaubte Werte .....	3712
6.18.169	chart.radiusAxis.axisPointer.lineStyle.shadowBlur .....	3712
6.18.170	chart.radiusAxis.axisPointer.lineStyle.shadowColor .....	3713
6.18.171	chart.radiusAxis.axisPointer.lineStyle.shadowOffsetX .....	3713
6.18.172	chart.radiusAxis.axisPointer.lineStyle.shadowOffsetY .....	3714

6.18.173	chart.radiusAxis.axisPointer.lineStyle.opacity.....	3714
6.18.174	chart.radiusAxis.axisPointer.shadowStyle.object.....	3714
6.18.175	chart.radiusAxis.axisPointer.shadowStyle.color.....	3715
6.18.176	chart.radiusAxis.axisPointer.shadowStyle.shadowBlur.....	3716
6.18.177	chart.radiusAxis.axisPointer.shadowStyle.shadowColor.....	3717
6.18.178	chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetX.....	3717
6.18.179	chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetY.....	3718
6.18.180	chart.radiusAxis.axisPointer.shadowStyle.opacity.....	3718
6.18.181	chart.radiusAxis.axisPointer.triggerTooltip.....	3718
6.18.182	chart.radiusAxis.axisPointer.value.....	3719
6.18.183	chart.radiusAxis.axisPointer.status.....	3719
	Erlaubte Werte.....	3719
6.18.184	chart.radiusAxis.axisPointer.handle.object.....	3719
6.18.185	chart.radiusAxis.axisPointer.handle.show.....	3720
6.18.186	chart.radiusAxis.axisPointer.handle.icon.....	3720
6.18.187	chart.radiusAxis.axisPointer.handle.size.....	3721
6.18.188	chart.radiusAxis.axisPointer.handle.margin.....	3721
6.18.189	chart.radiusAxis.axisPointer.handle.color.....	3721
6.18.190	chart.radiusAxis.axisPointer.handle.throttle.....	3722
6.18.191	chart.radiusAxis.axisPointer.handle.shadowBlur.....	3722
6.18.192	chart.radiusAxis.axisPointer.handle.shadowColor.....	3723
6.18.193	chart.radiusAxis.axisPointer.handle.shadowOffsetX.....	3723
6.18.194	chart.radiusAxis.axisPointer.handle.shadowOffsetY.....	3723
6.18.195	chart.radiusAxis.zlevel.....	3724
6.18.196	chart.radiusAxis.z.....	3724
6.18.197	chart.radiusAxis.nameTextStyle.borderType.....	3725
	Erlaubte Werte.....	3725
6.18.198	chart.radiusAxis.nameTextStyle.borderDashOffset.....	3725
6.18.199	chart.radiusAxis.axisLine.lineStyle.dashOffset.....	3725
6.18.200	chart.radiusAxis.axisLine.lineStyle.cap.....	3726
	Erlaubte Werte.....	3726
6.18.201	chart.radiusAxis.axisLine.lineStyle.join.....	3726
	Erlaubte Werte.....	3727
6.18.202	chart.radiusAxis.axisLine.lineStyle.miterLimit.....	3727
6.18.203	chart.radiusAxis.axisTick.lineStyle.dashOffset.....	3727
6.18.204	chart.radiusAxis.axisTick.lineStyle.cap.....	3728

	Erlaubte Werte .....	3728
6.18.205	chart.radiusAxis.axisTick.lineStyle.join .....	3728
	Erlaubte Werte .....	3729
6.18.206	chart.radiusAxis.splitLine.lineStyle.miterLimit .....	3729
6.18.207	chart.radiusAxis.splitLine.lineStyle.dashOffset .....	3729
6.18.208	chart.radiusAxis.splitLine.lineStyle.cap .....	3730
	Erlaubte Werte .....	3730
6.18.209	chart.radiusAxis.splitLine.lineStyle.join .....	3730
	Erlaubte Werte .....	3731
6.18.210	chart.radiusAxis.axisPointer.label.overflow .....	3731
	Erlaubte Werte .....	3731
6.18.211	chart.radiusAxis.axisPointer.label.ellipsis .....	3732
6.18.212	chart.radiusAxis.axisPointer.lineStyle.dashOffset .....	3732
6.18.213	chart.radiusAxis.axisPointer.lineStyle.cap .....	3732
	Erlaubte Werte .....	3733
6.18.214	chart.radiusAxis.axisPointer.lineStyle.join .....	3733
	Erlaubte Werte .....	3733
6.18.215	chart.radiusAxis.axisPointer.lineStyle.miterLimit .....	3733
6.18.216	chart.radiusAxis.minorTick.object .....	3734
6.18.217	chart.radiusAxis.minorTick.show .....	3734
6.18.218	chart.radiusAxis.minorTick.splitNumber .....	3735
6.18.219	chart.radiusAxis.minorTick.length .....	3735
6.18.220	chart.radiusAxis.minorTick.lineStyle.object .....	3735
6.18.221	chart.radiusAxis.minorTick.lineStyle.color .....	3736
6.18.222	chart.radiusAxis.minorTick.lineStyle.width .....	3736
6.18.223	chart.radiusAxis.minorTick.lineStyle.type .....	3736
	Erlaubte Werte .....	3737
6.18.224	chart.radiusAxis.minorTick.lineStyle.shadowBlur .....	3737
6.18.225	chart.radiusAxis.minorTick.lineStyle.shadowColor .....	3737
6.18.226	chart.radiusAxis.minorTick.lineStyle.shadowOffsetX .....	3738
6.18.227	chart.radiusAxis.minorTick.lineStyle.shadowOffsetY .....	3738
6.18.228	chart.radiusAxis.minorTick.lineStyle.opacity .....	3738
6.18.229	chart.radiusAxis.minorTick.lineStyle.dashOffset .....	3739
6.18.230	chart.radiusAxis.minorTick.lineStyle.cap .....	3739
	Erlaubte Werte .....	3740
6.18.231	chart.radiusAxis.minorTick.lineStyle.join .....	3740

	Erlaubte Werte .....	3740
6.18.232	chart.radiusAxis.minorTick.lineStyle.miterLimit .....	3740
6.18.233	chart.radiusAxis.minorSplitLine.object .....	3741
6.18.234	chart.radiusAxis.minorSplitLine.show .....	3741
6.18.235	chart.radiusAxis.minorSplitLine.lineStyle.object .....	3741
6.18.236	chart.radiusAxis.minorSplitLine.lineStyle.color .....	3742
6.18.237	chart.radiusAxis.minorSplitLine.lineStyle.width .....	3742
6.18.238	chart.radiusAxis.minorSplitLine.lineStyle.type.....	3742
	Erlaubte Werte .....	3743
6.18.239	chart.radiusAxis.minorSplitLine.lineStyle.dashOffset.....	3743
6.18.240	chart.radiusAxis.minorSplitLine.lineStyle.cap .....	3743
	Erlaubte Werte .....	3744
6.18.241	chart.radiusAxis.minorSplitLine.lineStyle.join.....	3744
	Erlaubte Werte .....	3745
6.18.242	chart.radiusAxis.minorSplitLine.lineStyle.miterLimit .....	3745
6.18.243	chart.radiusAxis.minorSplitLine.lineStyle.shadowBlur .....	3745
6.18.244	chart.radiusAxis.minorSplitLine.lineStyle.shadowColor .....	3746
6.18.245	chart.radiusAxis.minorSplitLine.lineStyle.shadowOffsetX.....	3746
6.18.246	chart.radiusAxis.minorSplitLine.lineStyle.shadowOffsetY .....	3746
6.18.247	chart.radiusAxis.minorSplitLine.lineStyle.opacity.....	3747
<b>6.19</b>	<b>ECharts (5) Series .....</b>	<b>3747</b>
6.19.1	chart.series.i[].activeOpacity.....	3747
6.19.2	chart.series.i[].animation .....	3747
6.19.3	chart.series.i[].animationDelay .....	3748
6.19.4	chart.series.i[].animationDelayUpdate .....	3748
6.19.5	chart.series.i[].animationDuration .....	3749
6.19.6	chart.series.i[].animationDurationUpdate .....	3749
6.19.7	chart.series.i[].animationEasing .....	3750
	Erlaubte Werte .....	3750
6.19.8	chart.series.i[].animationEasingUpdate .....	3751
	Erlaubte Werte .....	3751
6.19.9	chart.series.i[].animationEasingUpdate.animationDelay .....	3752
6.19.10	chart.series.i[].animationEasingUpdate.animationDelayUpdate.....	3752
6.19.11	chart.series.i[].animationEasingUpdate.object .....	3753
	Erlaubte Werte .....	3753
6.19.12	chart.series.i[].animationThreshold .....	3754

6.19.13	chart.series.i[].animationType .....	3754
	Erlaubte Werte .....	3755
6.19.14	chart.series.i[].areaStyle.color .....	3755
6.19.15	chart.series.i[].areaStyle.object .....	3756
6.19.16	chart.series.i[].areaStyle.opacity .....	3757
6.19.17	chart.series.i[].areaStyle.origin .....	3757
	Erlaubte Werte .....	3757
6.19.18	chart.series.i[].areaStyle.shadowBlur.....	3758
6.19.19	chart.series.i[].areaStyle.shadowColor .....	3758
6.19.20	chart.series.i[].areaStyle.shadowOffsetX .....	3758
6.19.21	chart.series.i[].areaStyle.shadowOffsetY.....	3759
6.19.22	chart.series.i[].aspectScale .....	3759
6.19.23	chart.series.i[].avoidLabelOverlap.....	3760
6.19.24	chart.series.i[].axisLabel.backgroundColor.....	3760
6.19.25	chart.series.i[].axisLabel.borderColor .....	3761
6.19.26	chart.series.i[].axisLabel.borderRadius .....	3761
6.19.27	chart.series.i[].axisLabel.borderWidth.....	3761
6.19.28	chart.series.i[].axisLabel.color .....	3762
6.19.29	chart.series.i[].axisLabel.fontFamily.....	3762
6.19.30	chart.series.i[].axisLabel.fontSize .....	3762
6.19.31	chart.series.i[].axisLabel.fontStyle.....	3763
	Erlaubte Werte .....	3763
6.19.32	chart.series.i[].axisLabel.fontWeight .....	3763
	Erlaubte Werte .....	3764
6.19.33	chart.series.i[].axisLabel.formatter.....	3764
6.19.34	chart.series.i[].axisLabel.height .....	3764
6.19.35	chart.series.i[].axisLabel.lineHeight .....	3765
6.19.36	chart.series.i[].axisLabel.object .....	3765
6.19.37	chart.series.i[].axisLabel.padding.....	3766
6.19.38	chart.series.i[].axisLabel.rich.object.....	3766
6.19.39	chart.series.i[].axisLabel.shadowBlur.....	3767
6.19.40	chart.series.i[].axisLabel.shadowColor.....	3768
6.19.41	chart.series.i[].axisLabel.shadowOffsetX.....	3768
6.19.42	chart.series.i[].axisLabel.shadowOffsetY.....	3768
6.19.43	chart.series.i[].axisLabel.show .....	3769
6.19.44	chart.series.i[].axisLabel.textBorderColor.....	3769

6.19.45	chart.series.i[].axisLabel.textBorderWidth .....	3769
6.19.46	chart.series.i[].axisLabel.textShadowBlur .....	3770
6.19.47	chart.series.i[].axisLabel.textShadowColor .....	3770
6.19.48	chart.series.i[].axisLabel.textShadowOffsetX .....	3770
6.19.49	chart.series.i[].axisLabel.textShadowOffsetY .....	3771
6.19.50	chart.series.i[].axisLabel.width .....	3771
6.19.51	chart.series.i[].axisLine.lineStyle.color .....	3772
6.19.52	chart.series.i[].axisLine.lineStyle.object .....	3772
6.19.53	chart.series.i[].axisLine.lineStyle.opacity .....	3772
6.19.54	chart.series.i[].axisLine.lineStyle.shadowBlur .....	3773
6.19.55	chart.series.i[].axisLine.lineStyle.shadowColor .....	3773
6.19.56	chart.series.i[].axisLine.lineStyle.shadowOffsetX .....	3774
6.19.57	chart.series.i[].axisLine.lineStyle.shadowOffsetY .....	3774
6.19.58	chart.series.i[].axisLine.lineStyle.width .....	3774
6.19.59	chart.series.i[].axisLine.object .....	3775
6.19.60	chart.series.i[].axisLine.show .....	3775
6.19.61	chart.series.i[].axisTick.length .....	3775
6.19.62	chart.series.i[].axisTick.lineStyle.color .....	3776
6.19.63	chart.series.i[].axisTick.lineStyle.object .....	3777
6.19.64	chart.series.i[].axisTick.lineStyle.opacity .....	3778
6.19.65	chart.series.i[].axisTick.lineStyle.shadowBlur .....	3778
6.19.66	chart.series.i[].axisTick.lineStyle.shadowColor .....	3779
6.19.67	chart.series.i[].axisTick.lineStyle.shadowOffsetX .....	3779
6.19.68	chart.series.i[].axisTick.lineStyle.shadowOffsetY .....	3779
6.19.69	chart.series.i[].axisTick.lineStyle.type .....	3780
	Erlaubte Werte .....	3780
6.19.70	chart.series.i[].axisTick.lineStyle.width .....	3780
6.19.71	chart.series.i[].axisTick.object .....	3780
6.19.72	chart.series.i[].axisTick.show .....	3781
6.19.73	chart.series.i[].axisTick.splitNumber .....	3781
6.19.74	chart.series.i[].barCategoryGap .....	3781
6.19.75	chart.series.i[].barGap .....	3782
6.19.76	chart.series.i[].barMaxWidth .....	3782
6.19.77	chart.series.i[].barMinHeight .....	3783
6.19.78	chart.series.i[].barMinWidth .....	3783
	Erlaubte Werte .....	3784



6.19.79	chart.series.i[].barWidth .....	3784
6.19.80	chart.series.i[].blurSize .....	3784
6.19.81	chart.series.i[].bottom .....	3784
	Erlaubte Werte .....	3785
6.19.82	chart.series.i[].boundaryGap.....	3785
	Erlaubte Werte .....	3785
6.19.83	chart.series.i[].boundingCoords .....	3785
6.19.84	chart.series.i[].boxWidth .....	3786
6.19.85	chart.series.i[].breadcrumb.bottom .....	3786
	Erlaubte Werte .....	3787
6.19.86	chart.series.i[].breadcrumb.emphasis.itemStyle.borderColor .....	3787
6.19.87	chart.series.i[].breadcrumb.emphasis.itemStyle.borderType .....	3787
	Erlaubte Werte .....	3788
6.19.88	chart.series.i[].breadcrumb.emphasis.itemStyle.borderWidth .....	3788
6.19.89	chart.series.i[].breadcrumb.emphasis.itemStyle.color .....	3788
6.19.90	chart.series.i[].breadcrumb.emphasis.itemStyle.object .....	3789
6.19.91	chart.series.i[].breadcrumb.emphasis.itemStyle.opacity .....	3790
6.19.92	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowBlur .....	3790
6.19.93	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowColor .....	3791
6.19.94	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetX .....	3791
6.19.95	chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetY .....	3791
6.19.96	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.align .....	3792
	Erlaubte Werte .....	3792
6.19.97	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.backgroundColor .....	3792
6.19.98	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderColor .....	3793
6.19.99	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderRadius.....	3793
6.19.100	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.borderWidth .....	3794
6.19.101	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.color .....	3794
6.19.102	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontFamily.....	3794
6.19.103	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontSize .....	3795
6.19.104	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontStyle .....	3795
	Erlaubte Werte .....	3795
6.19.105	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.fontWeight .....	3796
	Erlaubte Werte .....	3796
6.19.106	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.height .....	3796
6.19.107	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.lineHeight .....	3797

6.19.108	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.object .....	3797
6.19.109	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.padding.....	3798
6.19.110	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.rich.object.....	3798
6.19.111	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowBlur.....	3799
6.19.112	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowColor .....	3800
6.19.113	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetX .....	3800
6.19.114	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetY.....	3800
6.19.115	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textBorderColor.....	3801
6.19.116	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textBorderWidth.....	3801
6.19.117	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowBlur.....	3802
6.19.118	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowColor .....	3802
6.19.119	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetX .....	3802
6.19.120	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetY.....	3803
6.19.121	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.verticalAlign.....	3803
	Erlaubte Werte .....	3804
6.19.122	chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.width .....	3804
	Erlaubte Werte .....	3804
6.19.123	chart.series.i[].breadcrumb.emphasis.object .....	3805
6.19.124	chart.series.i[].breadcrumb.emptyItemWidth .....	3805
6.19.125	chart.series.i[].breadcrumb.height .....	3805
6.19.126	chart.series.i[].breadcrumb.itemStyle.borderColor .....	3806
6.19.127	chart.series.i[].breadcrumb.itemStyle.borderType .....	3806
	Erlaubte Werte .....	3806
6.19.128	chart.series.i[].breadcrumb.itemStyle.borderWidth .....	3806
6.19.129	chart.series.i[].breadcrumb.itemStyle.color .....	3807
6.19.130	chart.series.i[].breadcrumb.itemStyle.object .....	3808
6.19.131	chart.series.i[].breadcrumb.itemStyle.opacity .....	3809
6.19.132	chart.series.i[].breadcrumb.itemStyle.shadowBlur.....	3809
6.19.133	chart.series.i[].breadcrumb.itemStyle.shadowColor .....	3810
6.19.134	chart.series.i[].breadcrumb.itemStyle.shadowOffsetX .....	3810
6.19.135	chart.series.i[].breadcrumb.itemStyle.shadowOffsetY.....	3810
6.19.136	chart.series.i[].breadcrumb.itemStyle.textStyle.align.....	3811
	Erlaubte Werte .....	3811
6.19.137	chart.series.i[].breadcrumb.itemStyle.textStyle.backgroundColor .....	3811
6.19.138	chart.series.i[].breadcrumb.itemStyle.textStyle.borderColor .....	3812
6.19.139	chart.series.i[].breadcrumb.itemStyle.textStyle.borderRadius .....	3812

6.19.140	chart.series.i[].breadcrumb.itemStyle.textStyle.borderWidth .....	3813
6.19.141	chart.series.i[].breadcrumb.itemStyle.textStyle.color .....	3813
6.19.142	chart.series.i[].breadcrumb.itemStyle.textStyle.fontFamily.....	3813
6.19.143	chart.series.i[].breadcrumb.itemStyle.textStyle.fontSize .....	3814
6.19.144	chart.series.i[].breadcrumb.itemStyle.textStyle.fontStyle .....	3814
	Erlaubte Werte .....	3814
6.19.145	chart.series.i[].breadcrumb.itemStyle.textStyle.fontWeight .....	3815
	Erlaubte Werte .....	3815
6.19.146	chart.series.i[].breadcrumb.itemStyle.textStyle.height .....	3815
6.19.147	chart.series.i[].breadcrumb.itemStyle.textStyle.lineHeight .....	3816
6.19.148	chart.series.i[].breadcrumb.itemStyle.textStyle.object .....	3816
6.19.149	chart.series.i[].breadcrumb.itemStyle.textStyle.padding.....	3817
6.19.150	chart.series.i[].breadcrumb.itemStyle.textStyle.rich.object.....	3817
6.19.151	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowBlur.....	3818
6.19.152	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowColor.....	3819
6.19.153	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetX.....	3819
6.19.154	chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetY.....	3819
6.19.155	chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderColor.....	3820
6.19.156	chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderWidth.....	3820
6.19.157	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowBlur.....	3820
6.19.158	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowColor.....	3821
6.19.159	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetX.....	3821
6.19.160	chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetY.....	3821
6.19.161	chart.series.i[].breadcrumb.itemStyle.textStyle.verticalAlign .....	3822
	Erlaubte Werte .....	3822
6.19.162	chart.series.i[].breadcrumb.itemStyle.textStyle.width .....	3822
	Erlaubte Werte .....	3823
6.19.163	chart.series.i[].breadcrumb.left .....	3823
	Erlaubte Werte .....	3823
6.19.164	chart.series.i[].breadcrumb.object .....	3824
6.19.165	chart.series.i[].breadcrumb.right.....	3824
	Erlaubte Werte .....	3824
6.19.166	chart.series.i[].breadcrumb.show.....	3825
6.19.167	chart.series.i[].breadcrumb.top .....	3825
	Erlaubte Werte .....	3825
6.19.168	chart.series.i[].calendarIndex.....	3826

6.19.169	chart.series.i[].categories.emphasis.itemStyle.borderColor .....	3826
6.19.170	chart.series.i[].categories.emphasis.itemStyle.borderColor .....	3826
	Erlaubte Werte .....	3827
6.19.171	chart.series.i[].categories.emphasis.itemStyle.borderWidth.....	3827
6.19.172	chart.series.i[].categories.emphasis.itemStyle.color .....	3827
6.19.173	chart.series.i[].categories.emphasis.itemStyle.object .....	3828
6.19.174	chart.series.i[].categories.emphasis.itemStyle.opacity .....	3829
6.19.175	chart.series.i[].categories.emphasis.itemStyle.shadowBlur.....	3829
6.19.176	chart.series.i[].categories.emphasis.itemStyle.shadowColor.....	3830
6.19.177	chart.series.i[].categories.emphasis.itemStyle.shadowOffsetX.....	3830
6.19.178	chart.series.i[].categories.emphasis.itemStyle.shadowOffsetY.....	3830
6.19.179	chart.series.i[].categories.emphasis.label.align .....	3831
	Erlaubte Werte .....	3831
6.19.180	chart.series.i[].categories.emphasis.label.backgroundColor .....	3831
6.19.181	chart.series.i[].categories.emphasis.label.borderColor .....	3832
6.19.182	chart.series.i[].categories.emphasis.label.borderRadius.....	3832
6.19.183	chart.series.i[].categories.emphasis.label.borderWidth .....	3833
6.19.184	chart.series.i[].categories.emphasis.label.color .....	3833
6.19.185	chart.series.i[].categories.emphasis.label.distance .....	3833
6.19.186	chart.series.i[].categories.emphasis.label.fontFamily.....	3834
6.19.187	chart.series.i[].categories.emphasis.label.fontSize .....	3834
6.19.188	chart.series.i[].categories.emphasis.label.fontStyle .....	3835
	Erlaubte Werte .....	3835
6.19.189	chart.series.i[].categories.emphasis.label.fontWeight .....	3835
	Erlaubte Werte .....	3835
6.19.190	chart.series.i[].categories.emphasis.label.formatter .....	3836
6.19.191	chart.series.i[].categories.emphasis.label.height .....	3837
6.19.192	chart.series.i[].categories.emphasis.label.lineHeight .....	3837
6.19.193	chart.series.i[].categories.emphasis.label.object .....	3838
6.19.194	chart.series.i[].categories.emphasis.label.offset .....	3838
6.19.195	chart.series.i[].categories.emphasis.label.padding.....	3839
6.19.196	chart.series.i[].categories.emphasis.label.position .....	3839
	Erlaubte Werte .....	3840
6.19.197	chart.series.i[].categories.emphasis.label.rich.object.....	3840
6.19.198	chart.series.i[].categories.emphasis.label.rotate .....	3841
6.19.199	chart.series.i[].categories.emphasis.label.shadowBlur.....	3842

6.19.200	chart.series.i[].categories.emphasis.label.shadowColor .....	3842
6.19.201	chart.series.i[].categories.emphasis.label.shadowOffsetX .....	3842
6.19.202	chart.series.i[].categories.emphasis.label.shadowOffsetY .....	3843
6.19.203	chart.series.i[].categories.emphasis.label.show .....	3843
6.19.204	chart.series.i[].categories.emphasis.label.textBorderColor .....	3843
6.19.205	chart.series.i[].categories.emphasis.label.textBorderWidth .....	3844
6.19.206	chart.series.i[].categories.emphasis.label.textShadowBlur .....	3844
6.19.207	chart.series.i[].categories.emphasis.label.textShadowColor .....	3844
6.19.208	chart.series.i[].categories.emphasis.label.textShadowOffsetX .....	3845
6.19.209	chart.series.i[].categories.emphasis.label.textShadowOffsetY .....	3845
6.19.210	chart.series.i[].categories.emphasis.label.verticalAlign .....	3846
	Erlaubte Werte .....	3846
6.19.211	chart.series.i[].categories.emphasis.label.width .....	3846
6.19.212	chart.series.i[].categories.emphasis.object .....	3847
6.19.213	chart.series.i[].categories.itemStyle.borderColor .....	3847
6.19.214	chart.series.i[].categories.itemStyle.borderType .....	3847
	Erlaubte Werte .....	3848
6.19.215	chart.series.i[].categories.itemStyle.borderWidth .....	3848
6.19.216	chart.series.i[].categories.itemStyle.color .....	3848
6.19.217	chart.series.i[].categories.itemStyle.object .....	3850
6.19.218	chart.series.i[].categories.itemStyle.opacity .....	3850
6.19.219	chart.series.i[].categories.itemStyle.shadowBlur .....	3850
6.19.220	chart.series.i[].categories.itemStyle.shadowColor .....	3851
6.19.221	chart.series.i[].categories.itemStyle.shadowOffsetX .....	3851
6.19.222	chart.series.i[].categories.itemStyle.shadowOffsetY .....	3851
6.19.223	chart.series.i[].categories.label.align .....	3852
	Erlaubte Werte .....	3852
6.19.224	chart.series.i[].categories.label.backgroundColor .....	3852
6.19.225	chart.series.i[].categories.label.borderColor .....	3853
6.19.226	chart.series.i[].categories.label.borderRadius .....	3854
6.19.227	chart.series.i[].categories.label.borderWidth .....	3854
6.19.228	chart.series.i[].categories.label.color .....	3854
6.19.229	chart.series.i[].categories.label.distance .....	3855
6.19.230	chart.series.i[].categories.label.fontFamily .....	3855
6.19.231	chart.series.i[].categories.label.fontSize .....	3855
6.19.232	chart.series.i[].categories.label.fontStyle .....	3856

	Erlaubte Werte .....	3856
6.19.233	chart.series.i[].categories.label.fontWeight .....	3856
	Erlaubte Werte .....	3857
6.19.234	chart.series.i[].categories.label.formatter.....	3857
6.19.235	chart.series.i[].categories.label.height .....	3858
6.19.236	chart.series.i[].categories.label.lineHeight .....	3858
6.19.237	chart.series.i[].categories.label.object .....	3859
6.19.238	chart.series.i[].categories.label.offset .....	3859
6.19.239	chart.series.i[].categories.label.padding.....	3860
6.19.240	chart.series.i[].categories.label.position .....	3860
	Erlaubte Werte .....	3861
6.19.241	chart.series.i[].categories.label.rich.object.....	3861
6.19.242	chart.series.i[].categories.label.rotate.....	3862
6.19.243	chart.series.i[].categories.label.shadowBlur.....	3863
6.19.244	chart.series.i[].categories.label.shadowColor.....	3863
6.19.245	chart.series.i[].categories.label.shadowOffsetX.....	3863
6.19.246	chart.series.i[].categories.label.shadowOffsetY.....	3864
6.19.247	chart.series.i[].categories.label.show .....	3864
6.19.248	chart.series.i[].categories.label.textBorderColor.....	3864
6.19.249	chart.series.i[].categories.label.textBorderWidth.....	3865
6.19.250	chart.series.i[].categories.label.textShadowBlur.....	3865
6.19.251	chart.series.i[].categories.label.textShadowColor.....	3865
6.19.252	chart.series.i[].categories.label.textShadowOffsetX.....	3866
6.19.253	chart.series.i[].categories.label.textShadowOffsetY.....	3866
6.19.254	chart.series.i[].categories.label.verticalAlign .....	3867
	Erlaubte Werte .....	3867
6.19.255	chart.series.i[].categories.label.width .....	3867
6.19.256	chart.series.i[].categories.name.....	3868
6.19.257	chart.series.i[].categories.object.....	3868
6.19.258	chart.series.i[].categories.symbol.....	3868
	Erlaubte Werte .....	3869
6.19.259	chart.series.i[].categories.symbolKeepAspect .....	3869
6.19.260	chart.series.i[].categories.symbolOffset.....	3870
6.19.261	chart.series.i[].categories.symbolRotate.....	3870
6.19.262	chart.series.i[].categories.symbolSize .....	3870
6.19.263	chart.series.i[].center.....	3871

6.19.264	chart.series.i[].childrenVisibleMin.....	3871
6.19.265	chart.series.i[].circular.object .....	3872
6.19.266	chart.series.i[].circular.rotateLabel .....	3872
6.19.267	chart.series.i[].clipOverflow .....	3873
6.19.268	chart.series.i[].clockwise .....	3873
6.19.269	chart.series.i[].colorAlpha .....	3873
6.19.270	chart.series.i[].colorMappingBy .....	3874
	Erlaubte Werte .....	3875
6.19.271	chart.series.i[].colorSaturation .....	3875
6.19.272	chart.series.i[].connectNulls .....	3875
6.19.273	chart.series.i[].coordinateSystem.....	3876
	Erlaubte Werte .....	3876
6.19.274	chart.series.i[].cursor.....	3876
6.19.275	chart.series.i[].datasetIndex.....	3877
6.19.276	chart.series.i[].detail.backgroundColor.....	3877
6.19.277	chart.series.i[].detail.borderColor .....	3878
6.19.278	chart.series.i[].detail.borderRadius .....	3878
6.19.279	chart.series.i[].detail.borderWidth.....	3878
6.19.280	chart.series.i[].detail.color .....	3879
6.19.281	chart.series.i[].detail.fontFamily.....	3879
6.19.282	chart.series.i[].detail.fontSize .....	3879
6.19.283	chart.series.i[].detail.fontStyle.....	3880
	Erlaubte Werte .....	3880
6.19.284	chart.series.i[].detail.fontWeight .....	3880
	Erlaubte Werte .....	3881
6.19.285	chart.series.i[].detail.height .....	3881
6.19.286	chart.series.i[].detail.lineHeight.....	3881
6.19.287	chart.series.i[].detail.object .....	3882
6.19.288	chart.series.i[].detail.offsetCenter .....	3882
	Erlaubte Werte .....	3883
6.19.289	chart.series.i[].detail.padding.....	3883
6.19.290	chart.series.i[].detail.rich.object .....	3883
6.19.291	chart.series.i[].detail.shadowBlur.....	3884
6.19.292	chart.series.i[].detail.shadowColor.....	3885
6.19.293	chart.series.i[].detail.shadowOffsetX.....	3885
6.19.294	chart.series.i[].detail.shadowOffsetY.....	3885

6.19.295	chart.series.i[].detail.show .....	3886
6.19.296	chart.series.i[].detail.textBorderColor .....	3886
6.19.297	chart.series.i[].detail.textBorderWidth .....	3886
6.19.298	chart.series.i[].detail.textShadowBlur .....	3887
6.19.299	chart.series.i[].detail.textShadowColor .....	3887
6.19.300	chart.series.i[].detail.textShadowOffsetX .....	3887
6.19.301	chart.series.i[].detail.textShadowOffsetY .....	3888
6.19.302	chart.series.i[].detail.width .....	3888
6.19.303	chart.series.i[].dimensions .....	3889
6.19.304	chart.series.i[].downplay.itemStyle.borderColor .....	3890
6.19.305	chart.series.i[].downplay.itemStyle.borderType .....	3891
	Erlaubte Werte .....	3891
6.19.306	chart.series.i[].downplay.itemStyle.borderWidth.....	3891
6.19.307	chart.series.i[].downplay.itemStyle.color .....	3891
6.19.308	chart.series.i[].downplay.itemStyle.object .....	3893
6.19.309	chart.series.i[].downplay.itemStyle.opacity .....	3893
6.19.310	chart.series.i[].downplay.itemStyle.shadowBlur .....	3893
6.19.311	chart.series.i[].downplay.itemStyle.shadowColor .....	3894
6.19.312	chart.series.i[].downplay.itemStyle.shadowOffsetX.....	3894
6.19.313	chart.series.i[].downplay.itemStyle.shadowOffsetY.....	3895
6.19.314	chart.series.i[].downplay.label.align .....	3895
	Erlaubte Werte .....	3895
6.19.315	chart.series.i[].downplay.label.backgroundColor .....	3895
6.19.316	chart.series.i[].downplay.label.borderColor .....	3896
6.19.317	chart.series.i[].downplay.label.borderRadius.....	3897
6.19.318	chart.series.i[].downplay.label.borderWidth .....	3897
6.19.319	chart.series.i[].downplay.label.color .....	3897
6.19.320	chart.series.i[].downplay.label.distance .....	3898
6.19.321	chart.series.i[].downplay.label.fontFamily.....	3898
6.19.322	chart.series.i[].downplay.label.fontSize .....	3898
6.19.323	chart.series.i[].downplay.label.fontStyle .....	3899
	Erlaubte Werte .....	3899
6.19.324	chart.series.i[].downplay.label.fontWeight .....	3899
	Erlaubte Werte .....	3900
6.19.325	chart.series.i[].downplay.label.formatter .....	3900
6.19.326	chart.series.i[].downplay.label.height.....	3901



6.19.327	chart.series.i[].downplay.label.lineHeight .....	3901
6.19.328	chart.series.i[].downplay.label.minAngle.....	3902
6.19.329	chart.series.i[].downplay.label.object .....	3902
6.19.330	chart.series.i[].downplay.label.offset .....	3903
6.19.331	chart.series.i[].downplay.label.padding.....	3903
6.19.332	chart.series.i[].downplay.label.position .....	3904
	Erlaubte Werte .....	3904
6.19.333	chart.series.i[].downplay.label.rich.object.....	3904
6.19.334	chart.series.i[].downplay.label.rotate .....	3905
	Erlaubte Werte .....	3906
6.19.335	chart.series.i[].downplay.label.shadowBlur .....	3906
6.19.336	chart.series.i[].downplay.label.shadowColor .....	3906
6.19.337	chart.series.i[].downplay.label.shadowOffsetX .....	3907
6.19.338	chart.series.i[].downplay.label.shadowOffsetY .....	3907
6.19.339	chart.series.i[].downplay.label.show.....	3907
6.19.340	chart.series.i[].downplay.label.textBorderColor.....	3908
6.19.341	chart.series.i[].downplay.label.textBorderWidth.....	3908
6.19.342	chart.series.i[].downplay.label.textShadowBlur.....	3909
6.19.343	chart.series.i[].downplay.label.textShadowColor .....	3909
6.19.344	chart.series.i[].downplay.label.textShadowOffsetX .....	3909
6.19.345	chart.series.i[].downplay.label.textShadowOffsetY.....	3910
6.19.346	chart.series.i[].downplay.label.verticalAlign.....	3910
	Erlaubte Werte .....	3911
6.19.347	chart.series.i[].downplay.label.width .....	3911
6.19.348	chart.series.i[].downplay.object .....	3911
6.19.349	chart.series.i[].draggable.....	3912
6.19.350	chart.series.i[].drillDownIcon .....	3912
6.19.351	chart.series.i[].edgeLabel.align .....	3912
	Erlaubte Werte .....	3913
6.19.352	chart.series.i[].edgeLabel.backgroundColor.....	3913
6.19.353	chart.series.i[].edgeLabel.borderColor.....	3914
6.19.354	chart.series.i[].edgeLabel.borderRadius .....	3914
6.19.355	chart.series.i[].edgeLabel.borderWidth.....	3914
6.19.356	chart.series.i[].edgeLabel.color.....	3915
6.19.357	chart.series.i[].edgeLabel.fontFamily .....	3915
6.19.358	chart.series.i[].edgeLabel.fontSize .....	3915

6.19.359	chart.series.i[].edgeLabel.fontStyle .....	3916
	Erlaubte Werte .....	3916
6.19.360	chart.series.i[].edgeLabel.fontWeight .....	3916
	Erlaubte Werte .....	3917
6.19.361	chart.series.i[].edgeLabel.formatter .....	3917
6.19.362	chart.series.i[].edgeLabel.height .....	3918
6.19.363	chart.series.i[].edgeLabel.lineHeight .....	3918
6.19.364	chart.series.i[].edgeLabel.object .....	3919
6.19.365	chart.series.i[].edgeLabel.padding .....	3919
6.19.366	chart.series.i[].edgeLabel.position .....	3920
	Erlaubte Werte .....	3920
6.19.367	chart.series.i[].edgeLabel.rich.object .....	3920
6.19.368	chart.series.i[].edgeLabel.shadowBlur .....	3921
6.19.369	chart.series.i[].edgeLabel.shadowColor .....	3922
6.19.370	chart.series.i[].edgeLabel.shadowOffsetX .....	3922
6.19.371	chart.series.i[].edgeLabel.shadowOffsetY .....	3922
6.19.372	chart.series.i[].edgeLabel.show .....	3923
6.19.373	chart.series.i[].edgeLabel.textBorderColor .....	3923
6.19.374	chart.series.i[].edgeLabel.textBorderWidth .....	3923
6.19.375	chart.series.i[].edgeLabel.textShadowBlur .....	3924
6.19.376	chart.series.i[].edgeLabel.textShadowColor .....	3924
6.19.377	chart.series.i[].edgeLabel.textShadowOffsetX .....	3924
6.19.378	chart.series.i[].edgeLabel.textShadowOffsetY .....	3925
6.19.379	chart.series.i[].edgeLabel.verticalAlign .....	3925
	Erlaubte Werte .....	3926
6.19.380	chart.series.i[].edgeLabel.width .....	3926
6.19.381	chart.series.i[].edges .....	3926
6.19.382	chart.series.i[].edgeSymbol .....	3927
	Erlaubte Werte .....	3927
6.19.383	chart.series.i[].edgeSymbolSize .....	3927
6.19.384	chart.series.i[].effect.color .....	3928
6.19.385	chart.series.i[].effect.constantSpeed .....	3928
6.19.386	chart.series.i[].effect.delay .....	3929
6.19.387	chart.series.i[].effect.loop .....	3929
6.19.388	chart.series.i[].effect.object .....	3929
6.19.389	chart.series.i[].effect.period .....	3930

6.19.390	chart.series.i[].effect.show .....	3930
6.19.391	chart.series.i[].effect.symbol.....	3930
	Erlaubte Werte .....	3931
6.19.392	chart.series.i[].effect.symbolSize .....	3931
6.19.393	chart.series.i[].effect.trailLength.....	3932
6.19.394	chart.series.i[].effectType.....	3932
	Erlaubte Werte .....	3932
6.19.395	chart.series.i[].emphasis.edgeLabel.align.....	3932
	Erlaubte Werte .....	3933
6.19.396	chart.series.i[].emphasis.edgeLabel.backgroundColor.....	3933
6.19.397	chart.series.i[].emphasis.edgeLabel.borderColor.....	3934
6.19.398	chart.series.i[].emphasis.edgeLabel.borderRadius .....	3934
6.19.399	chart.series.i[].emphasis.edgeLabel.borderWidth.....	3935
6.19.400	chart.series.i[].emphasis.edgeLabel.color .....	3935
6.19.401	chart.series.i[].emphasis.edgeLabel.fontFamily .....	3935
6.19.402	chart.series.i[].emphasis.edgeLabel.fontSize .....	3936
6.19.403	chart.series.i[].emphasis.edgeLabel.fontStyle.....	3936
	Erlaubte Werte .....	3936
6.19.404	chart.series.i[].emphasis.edgeLabel.fontWeight .....	3936
	Erlaubte Werte .....	3937
6.19.405	chart.series.i[].emphasis.edgeLabel.formatter .....	3937
6.19.406	chart.series.i[].emphasis.edgeLabel.height .....	3938
6.19.407	chart.series.i[].emphasis.edgeLabel.lineHeight.....	3939
6.19.408	chart.series.i[].emphasis.edgeLabel.object .....	3939
6.19.409	chart.series.i[].emphasis.edgeLabel.padding .....	3940
6.19.410	chart.series.i[].emphasis.edgeLabel.position .....	3940
	Erlaubte Werte .....	3940
6.19.411	chart.series.i[].emphasis.edgeLabel.rich.object .....	3940
6.19.412	chart.series.i[].emphasis.edgeLabel.shadowBlur .....	3941
6.19.413	chart.series.i[].emphasis.edgeLabel.shadowColor .....	3942
6.19.414	chart.series.i[].emphasis.edgeLabel.shadowOffsetX.....	3942
6.19.415	chart.series.i[].emphasis.edgeLabel.shadowOffsetY.....	3943
6.19.416	chart.series.i[].emphasis.edgeLabel.show .....	3943
6.19.417	chart.series.i[].emphasis.edgeLabel.textBorderColor .....	3943
6.19.418	chart.series.i[].emphasis.edgeLabel.textBorderWidth .....	3944
6.19.419	chart.series.i[].emphasis.edgeLabel.textShadowBlur .....	3944

6.19.420	chart.series.i[].emphasis.edgeLabel.textShadowColor .....	3944
6.19.421	chart.series.i[].emphasis.edgeLabel.textShadowOffsetX .....	3945
6.19.422	chart.series.i[].emphasis.edgeLabel.textShadowOffsetY .....	3945
6.19.423	chart.series.i[].emphasis.edgeLabel.verticalAlign .....	3945
	Erlaubte Werte .....	3946
6.19.424	chart.series.i[].emphasis.edgeLabel.width .....	3946
6.19.425	chart.series.i[].emphasis.itemStyle.borderColor .....	3947
6.19.426	chart.series.i[].emphasis.itemStyle.borderType .....	3947
	Erlaubte Werte .....	3947
6.19.427	chart.series.i[].emphasis.itemStyle.borderWidth .....	3947
6.19.428	chart.series.i[].emphasis.itemStyle.color .....	3948
6.19.429	chart.series.i[].emphasis.itemStyle.object .....	3949
6.19.430	chart.series.i[].emphasis.itemStyle.opacity .....	3950
6.19.431	chart.series.i[].emphasis.itemStyle.shadowBlur .....	3950
6.19.432	chart.series.i[].emphasis.itemStyle.shadowColor .....	3951
6.19.433	chart.series.i[].emphasis.itemStyle.shadowOffsetX .....	3951
6.19.434	chart.series.i[].emphasis.itemStyle.shadowOffsetY .....	3951
6.19.435	chart.series.i[].emphasis.label.align .....	3952
	Erlaubte Werte .....	3952
6.19.436	chart.series.i[].emphasis.label.backgroundColor .....	3952
6.19.437	chart.series.i[].emphasis.label.borderColor .....	3953
6.19.438	chart.series.i[].emphasis.label.borderRadius .....	3953
6.19.439	chart.series.i[].emphasis.label.borderWidth .....	3954
6.19.440	chart.series.i[].emphasis.label.color .....	3954
6.19.441	chart.series.i[].emphasis.label.distance .....	3954
6.19.442	chart.series.i[].emphasis.label.fontFamily .....	3955
6.19.443	chart.series.i[].emphasis.label.fontSize .....	3955
6.19.444	chart.series.i[].emphasis.label.fontStyle .....	3956
	Erlaubte Werte .....	3956
6.19.445	chart.series.i[].emphasis.label.fontWeight .....	3956
	Erlaubte Werte .....	3956
6.19.446	chart.series.i[].emphasis.label.formatter .....	3957
6.19.447	chart.series.i[].emphasis.label.height .....	3958
6.19.448	chart.series.i[].emphasis.label.lineHeight .....	3958
6.19.449	chart.series.i[].emphasis.label.minAngle .....	3959
6.19.450	chart.series.i[].emphasis.label.object .....	3959

6.19.451	chart.series.i[].emphasis.label.offset.....	3960
6.19.452	chart.series.i[].emphasis.label.padding .....	3960
6.19.453	chart.series.i[].emphasis.label.position .....	3960
	Erlaubte Werte .....	3961
6.19.454	chart.series.i[].emphasis.label.rich.object .....	3961
6.19.455	chart.series.i[].emphasis.label.rotate .....	3962
6.19.456	chart.series.i[].emphasis.label.shadowBlur .....	3963
6.19.457	chart.series.i[].emphasis.label.shadowColor .....	3963
6.19.458	chart.series.i[].emphasis.label.shadowOffsetX .....	3963
6.19.459	chart.series.i[].emphasis.label.shadowOffsetY .....	3964
6.19.460	chart.series.i[].emphasis.label.show .....	3964
6.19.461	chart.series.i[].emphasis.label.textBorderColor .....	3964
6.19.462	chart.series.i[].emphasis.label.textBorderWidth .....	3965
6.19.463	chart.series.i[].emphasis.label.textShadowBlur .....	3965
6.19.464	chart.series.i[].emphasis.label.textShadowColor .....	3966
6.19.465	chart.series.i[].emphasis.label.textShadowOffsetX .....	3966
6.19.466	chart.series.i[].emphasis.label.textShadowOffsetY .....	3966
6.19.467	chart.series.i[].emphasis.label.verticalAlign .....	3967
	Erlaubte Werte .....	3967
6.19.468	chart.series.i[].emphasis.label.width.....	3967
6.19.469	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.color.....	3968
6.19.470	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.object.....	3969
6.19.471	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.opacity.....	3970
6.19.472	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowBlur .....	3970
6.19.473	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowColor .....	3971
6.19.474	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetX .....	3971
6.19.475	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetY .....	3971
6.19.476	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.type.....	3972
	Erlaubte Werte .....	3972
6.19.477	chart.series.i[].emphasis.labelLine.emphasis.lineStyle.width.....	3972
6.19.478	chart.series.i[].emphasis.labelLine.emphasis.object .....	3972
6.19.479	chart.series.i[].emphasis.labelLine.emphasis.show .....	3973
6.19.480	chart.series.i[].emphasis.labelLine.length .....	3973
6.19.481	chart.series.i[].emphasis.labelLine.lineStyle.color .....	3973
6.19.482	chart.series.i[].emphasis.labelLine.lineStyle.object.....	3975
6.19.483	chart.series.i[].emphasis.labelLine.lineStyle.opacity.....	3976

6.19.484	chart.series.i[].emphasis.labelLine.lineStyle.shadowBlur .....	3976
6.19.485	chart.series.i[].emphasis.labelLine.lineStyle.shadowColor .....	3977
6.19.486	chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetX .....	3977
6.19.487	chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetY .....	3977
6.19.488	chart.series.i[].emphasis.labelLine.lineStyle.type .....	3978
	Erlaubte Werte .....	3978
6.19.489	chart.series.i[].emphasis.labelLine.lineStyle.width .....	3978
6.19.490	chart.series.i[].emphasis.labelLine.object .....	3978
6.19.491	chart.series.i[].emphasis.labelLine.show .....	3979
6.19.492	chart.series.i[].emphasis.lineStyle.color .....	3979
6.19.493	chart.series.i[].emphasis.lineStyle.curveness .....	3980
6.19.494	chart.series.i[].emphasis.lineStyle.object .....	3981
6.19.495	chart.series.i[].emphasis.lineStyle.shadowBlur .....	3981
6.19.496	chart.series.i[].emphasis.lineStyle.shadowColor .....	3982
6.19.497	chart.series.i[].emphasis.lineStyle.shadowOffsetX .....	3982
6.19.498	chart.series.i[].emphasis.lineStyle.shadowOffsetY .....	3982
6.19.499	chart.series.i[].emphasis.lineStyle.width .....	3983
6.19.500	chart.series.i[].emphasis.object .....	3983
6.19.501	chart.series.i[].emphasis.upperLabel.align .....	3983
	Erlaubte Werte .....	3984
6.19.502	chart.series.i[].emphasis.upperLabel.backgroundColor .....	3984
6.19.503	chart.series.i[].emphasis.upperLabel.borderColor .....	3985
6.19.504	chart.series.i[].emphasis.upperLabel.borderRadius .....	3985
6.19.505	chart.series.i[].emphasis.upperLabel.borderWidth .....	3985
6.19.506	chart.series.i[].emphasis.upperLabel.color .....	3986
6.19.507	chart.series.i[].emphasis.upperLabel.distance .....	3986
6.19.508	chart.series.i[].emphasis.upperLabel.ellipsis .....	3987
6.19.509	chart.series.i[].emphasis.upperLabel.fontFamily .....	3987
6.19.510	chart.series.i[].emphasis.upperLabel.fontSize .....	3987
6.19.511	chart.series.i[].emphasis.upperLabel.fontStyle .....	3988
	Erlaubte Werte .....	3988
6.19.512	chart.series.i[].emphasis.upperLabel.fontWeight .....	3988
	Erlaubte Werte .....	3989
6.19.513	chart.series.i[].emphasis.upperLabel.formatter .....	3989
6.19.514	chart.series.i[].emphasis.upperLabel.height .....	3990
6.19.515	chart.series.i[].emphasis.upperLabel.lineHeight .....	3990

6.19.516	chart.series.i[].emphasis.upperLabel.object.....	3991
6.19.517	chart.series.i[].emphasis.upperLabel.offset.....	3991
6.19.518	chart.series.i[].emphasis.upperLabel.padding .....	3992
6.19.519	chart.series.i[].emphasis.upperLabel.position .....	3992
	Erlaubte Werte .....	3993
6.19.520	chart.series.i[].emphasis.upperLabel.rich.object .....	3993
6.19.521	chart.series.i[].emphasis.upperLabel.rotate .....	3994
6.19.522	chart.series.i[].emphasis.upperLabel.shadowBlur .....	3995
6.19.523	chart.series.i[].emphasis.upperLabel.shadowColor .....	3995
6.19.524	chart.series.i[].emphasis.upperLabel.shadowOffsetX .....	3995
6.19.525	chart.series.i[].emphasis.upperLabel.shadowOffsetY .....	3996
6.19.526	chart.series.i[].emphasis.upperLabel.show .....	3996
6.19.527	chart.series.i[].emphasis.upperLabel.textBorderColor .....	3996
6.19.528	chart.series.i[].emphasis.upperLabel.textBorderWidth .....	3997
6.19.529	chart.series.i[].emphasis.upperLabel.textShadowBlur .....	3997
6.19.530	chart.series.i[].emphasis.upperLabel.textShadowColor .....	3997
6.19.531	chart.series.i[].emphasis.upperLabel.textShadowOffsetX .....	3998
6.19.532	chart.series.i[].emphasis.upperLabel.textShadowOffsetY .....	3998
6.19.533	chart.series.i[].emphasis.upperLabel.verticalAlign .....	3999
	Erlaubte Werte .....	3999
6.19.534	chart.series.i[].emphasis.upperLabel.width.....	3999
6.19.535	chart.series.i[].encode .....	4000
6.19.536	chart.series.i[].endAngle.....	4002
6.19.537	chart.series.i[].expandAndCollapse .....	4002
6.19.538	chart.series.i[].focusNodeAdjacency .....	4002
	Erlaubte Werte .....	4003
6.19.539	chart.series.i[].force.edgeLength .....	4003
6.19.540	chart.series.i[].force.gravity .....	4004
6.19.541	chart.series.i[].force.initLayout.....	4004
6.19.542	chart.series.i[].force.layoutAnimation .....	4004
6.19.543	chart.series.i[].force.object .....	4005
6.19.544	chart.series.i[].force.repulsion .....	4005
6.19.545	chart.series.i[].funnelAlign .....	4006
	Erlaubte Werte .....	4006
6.19.546	chart.series.i[].gap .....	4006
6.19.547	chart.series.i[].geolIndex .....	4006

6.19.548	chart.series.i[].height.....	4007
6.19.549	chart.series.i[].highlight.itemStyle.borderColor.....	4007
6.19.550	chart.series.i[].highlight.itemStyle.borderType.....	4008
	Erlaubte Werte .....	4008
6.19.551	chart.series.i[].highlight.itemStyle.borderWidth.....	4008
6.19.552	chart.series.i[].highlight.itemStyle.color.....	4008
6.19.553	chart.series.i[].highlight.itemStyle.object.....	4010
6.19.554	chart.series.i[].highlight.itemStyle.opacity.....	4010
6.19.555	chart.series.i[].highlight.itemStyle.shadowBlur.....	4010
6.19.556	chart.series.i[].highlight.itemStyle.shadowColor.....	4011
6.19.557	chart.series.i[].highlight.itemStyle.shadowOffsetX.....	4011
6.19.558	chart.series.i[].highlight.itemStyle.shadowOffsetY.....	4012
6.19.559	chart.series.i[].highlight.label.align.....	4012
	Erlaubte Werte .....	4012
6.19.560	chart.series.i[].highlight.label.backgroundColor.....	4012
6.19.561	chart.series.i[].highlight.label.borderColor.....	4013
6.19.562	chart.series.i[].highlight.label.borderRadius.....	4014
6.19.563	chart.series.i[].highlight.label.borderWidth.....	4014
6.19.564	chart.series.i[].highlight.label.color.....	4014
6.19.565	chart.series.i[].highlight.label.distance.....	4015
6.19.566	chart.series.i[].highlight.label.fontFamily.....	4015
6.19.567	chart.series.i[].highlight.label.fontSize.....	4015
6.19.568	chart.series.i[].highlight.label.fontStyle.....	4016
	Erlaubte Werte .....	4016
6.19.569	chart.series.i[].highlight.label.fontWeight.....	4016
	Erlaubte Werte .....	4017
6.19.570	chart.series.i[].highlight.label.formatter.....	4017
6.19.571	chart.series.i[].highlight.label.height.....	4018
6.19.572	chart.series.i[].highlight.label.lineHeight.....	4018
6.19.573	chart.series.i[].highlight.label.minAngle.....	4019
6.19.574	chart.series.i[].highlight.label.object.....	4019
6.19.575	chart.series.i[].highlight.label.offset.....	4020
6.19.576	chart.series.i[].highlight.label.padding.....	4020
6.19.577	chart.series.i[].highlight.label.position.....	4021
	Erlaubte Werte .....	4021
6.19.578	chart.series.i[].highlight.label.rich.object.....	4021



6.19.579	chart.series.i[].highlight.label.rotate .....	4022
	Erlaubte Werte .....	4023
6.19.580	chart.series.i[].highlight.label.shadowBlur .....	4023
6.19.581	chart.series.i[].highlight.label.shadowColor .....	4023
6.19.582	chart.series.i[].highlight.label.shadowOffsetX .....	4024
6.19.583	chart.series.i[].highlight.label.shadowOffsetY .....	4024
6.19.584	chart.series.i[].highlight.label.show .....	4024
6.19.585	chart.series.i[].highlight.label.textBorderColor .....	4025
6.19.586	chart.series.i[].highlight.label.textBorderWidth .....	4025
6.19.587	chart.series.i[].highlight.label.textShadowBlur .....	4026
6.19.588	chart.series.i[].highlight.label.textShadowColor .....	4026
6.19.589	chart.series.i[].highlight.label.textShadowOffsetX .....	4026
6.19.590	chart.series.i[].highlight.label.textShadowOffsetY .....	4027
6.19.591	chart.series.i[].highlight.label.verticalAlign .....	4027
	Erlaubte Werte .....	4028
6.19.592	chart.series.i[].highlight.label.width.....	4028
6.19.593	chart.series.i[].highlight.object .....	4028
6.19.594	chart.series.i[].highlightPolicy .....	4029
	Erlaubte Werte .....	4029
6.19.595	chart.series.i[].hoverAnimation .....	4029
6.19.596	chart.series.i[].hoverAnimation.animation .....	4030
6.19.597	chart.series.i[].hoverAnimation.animationDelay.....	4030
6.19.598	chart.series.i[].hoverAnimation.animationDelayUpdate .....	4031
6.19.599	chart.series.i[].hoverAnimation.animationDuration .....	4031
6.19.600	chart.series.i[].hoverAnimation.animationDurationUpdate .....	4032
6.19.601	chart.series.i[].hoverAnimation.animationEasing .....	4032
	Erlaubte Werte .....	4033
6.19.602	chart.series.i[].hoverAnimation.animationEasingUpdate.....	4033
	Erlaubte Werte .....	4034
6.19.603	chart.series.i[].hoverAnimation.animationThreshold .....	4034
6.19.604	chart.series.i[].hoverAnimation.object .....	4035
6.19.605	chart.series.i[].hoverOffset .....	4035
6.19.606	chart.series.i[].id .....	4036
6.19.607	chart.series.i[].inactiveOpacity .....	4036
6.19.608	chart.series.i[].initialTreeDepth .....	4037
6.19.609	chart.series.i[].itemStyle.borderColor .....	4037

6.19.610	chart.series.i[].itemStyle.borderColor	4037
	Erlaubte Werte	4038
6.19.611	chart.series.i[].itemStyle.borderWidth	4038
6.19.612	chart.series.i[].itemStyle.color	4038
6.19.613	chart.series.i[].itemStyle.object	4039
6.19.614	chart.series.i[].itemStyle.opacity	4040
6.19.615	chart.series.i[].itemStyle.shadowBlur	4040
6.19.616	chart.series.i[].itemStyle.shadowColor	4041
6.19.617	chart.series.i[].itemStyle.shadowOffsetX	4041
6.19.618	chart.series.i[].itemStyle.shadowOffsetY	4041
6.19.619	chart.series.i[].label.align	4042
	Erlaubte Werte	4042
6.19.620	chart.series.i[].label.backgroundColor	4042
6.19.621	chart.series.i[].label.borderColor	4043
6.19.622	chart.series.i[].label.borderRadius	4043
6.19.623	chart.series.i[].label.borderWidth	4044
6.19.624	chart.series.i[].label.color	4044
6.19.625	chart.series.i[].label.distance	4044
6.19.626	chart.series.i[].label.fontFamily	4045
6.19.627	chart.series.i[].label.fontSize	4045
6.19.628	chart.series.i[].label.fontStyle	4046
	Erlaubte Werte	4046
6.19.629	chart.series.i[].label.fontWeight	4046
	Erlaubte Werte	4046
6.19.630	chart.series.i[].label.formatter	4047
6.19.631	chart.series.i[].label.height	4048
6.19.632	chart.series.i[].label.lineHeight	4048
6.19.633	chart.series.i[].label.minAngle	4049
6.19.634	chart.series.i[].label.object	4049
6.19.635	chart.series.i[].label.offset	4050
6.19.636	chart.series.i[].label.padding	4050
6.19.637	chart.series.i[].label.position	4051
	Erlaubte Werte	4051
6.19.638	chart.series.i[].label.rich.object	4051
6.19.639	chart.series.i[].label.rotate	4052
6.19.640	chart.series.i[].label.shadowBlur	4053

6.19.641	chart.series.i[].label.shadowColor .....	4053
6.19.642	chart.series.i[].label.shadowOffsetX .....	4053
6.19.643	chart.series.i[].label.shadowOffsetY .....	4054
6.19.644	chart.series.i[].label.show .....	4054
6.19.645	chart.series.i[].label.textBorderColor .....	4054
6.19.646	chart.series.i[].label.textBorderWidth .....	4055
6.19.647	chart.series.i[].label.textShadowBlur .....	4055
6.19.648	chart.series.i[].label.textShadowColor .....	4056
6.19.649	chart.series.i[].label.textShadowOffsetX .....	4056
6.19.650	chart.series.i[].label.textShadowOffsetY .....	4056
6.19.651	chart.series.i[].label.verticalAlign .....	4057
	Erlaubte Werte .....	4057
6.19.652	chart.series.i[].label.width.....	4057
6.19.653	chart.series.i[].labelLine.emphasis.lineStyle.color .....	4058
6.19.654	chart.series.i[].labelLine.emphasis.lineStyle.object .....	4059
6.19.655	chart.series.i[].labelLine.emphasis.lineStyle.opacity .....	4060
6.19.656	chart.series.i[].labelLine.emphasis.lineStyle.shadowBlur .....	4060
6.19.657	chart.series.i[].labelLine.emphasis.lineStyle.shadowColor .....	4061
6.19.658	chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetX .....	4061
6.19.659	chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetY .....	4061
6.19.660	chart.series.i[].labelLine.emphasis.lineStyle.type .....	4062
	Erlaubte Werte .....	4062
6.19.661	chart.series.i[].labelLine.emphasis.lineStyle.width.....	4062
6.19.662	chart.series.i[].labelLine.emphasis.object .....	4062
6.19.663	chart.series.i[].labelLine.emphasis.show .....	4063
6.19.664	chart.series.i[].labelLine.length .....	4063
6.19.665	chart.series.i[].labelLine.length2 .....	4063
6.19.666	chart.series.i[].labelLine.lineStyle.color .....	4064
6.19.667	chart.series.i[].labelLine.lineStyle.object .....	4065
6.19.668	chart.series.i[].labelLine.lineStyle.opacity .....	4066
6.19.669	chart.series.i[].labelLine.lineStyle.shadowBlur .....	4066
6.19.670	chart.series.i[].labelLine.lineStyle.shadowColor .....	4067
6.19.671	chart.series.i[].labelLine.lineStyle.shadowOffsetX .....	4067
6.19.672	chart.series.i[].labelLine.lineStyle.shadowOffsetY .....	4067
6.19.673	chart.series.i[].labelLine.lineStyle.type .....	4068
	Erlaubte Werte .....	4068

6.19.674	chart.series.i[].labelLine.lineStyle.width .....	4068
6.19.675	chart.series.i[].labelLine.object .....	4068
6.19.676	chart.series.i[].labelLine.show .....	4069
6.19.677	chart.series.i[].labelLine.smooth .....	4069
6.19.678	chart.series.i[].large .....	4069
6.19.679	chart.series.i[].largeThreshold .....	4070
6.19.680	chart.series.i[].layout .....	4070
6.19.681	chart.series.i[].layoutIterations .....	4071
6.19.682	chart.series.i[].leafDepth .....	4071
6.19.683	chart.series.i[].leaves.emphasis.itemStyle.borderColor .....	4072
6.19.684	chart.series.i[].leaves.emphasis.itemStyle.borderType .....	4072
	Erlaubte Werte .....	4073
6.19.685	chart.series.i[].leaves.emphasis.itemStyle.borderWidth .....	4073
6.19.686	chart.series.i[].leaves.emphasis.itemStyle.color .....	4073
6.19.687	chart.series.i[].leaves.emphasis.itemStyle.object .....	4074
6.19.688	chart.series.i[].leaves.emphasis.itemStyle.opacity .....	4075
6.19.689	chart.series.i[].leaves.emphasis.itemStyle.shadowBlur .....	4075
6.19.690	chart.series.i[].leaves.emphasis.itemStyle.shadowColor .....	4076
6.19.691	chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetX .....	4076
6.19.692	chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetY .....	4076
6.19.693	chart.series.i[].leaves.emphasis.label.align .....	4077
	Erlaubte Werte .....	4077
6.19.694	chart.series.i[].leaves.emphasis.label.backgroundColor .....	4077
6.19.695	chart.series.i[].leaves.emphasis.label.borderColor .....	4078
6.19.696	chart.series.i[].leaves.emphasis.label.borderRadius .....	4078
6.19.697	chart.series.i[].leaves.emphasis.label.borderWidth .....	4079
6.19.698	chart.series.i[].leaves.emphasis.label.color .....	4079
6.19.699	chart.series.i[].leaves.emphasis.label.distance .....	4079
6.19.700	chart.series.i[].leaves.emphasis.label.fontFamily .....	4080
6.19.701	chart.series.i[].leaves.emphasis.label.fontSize .....	4080
6.19.702	chart.series.i[].leaves.emphasis.label.fontStyle .....	4081
	Erlaubte Werte .....	4081
6.19.703	chart.series.i[].leaves.emphasis.label.fontWeight .....	4081
	Erlaubte Werte .....	4081
6.19.704	chart.series.i[].leaves.emphasis.label.height .....	4082
6.19.705	chart.series.i[].leaves.emphasis.label.lineHeight .....	4082

6.19.706	chart.series.i[].leaves.emphasis.label.object .....	4083
6.19.707	chart.series.i[].leaves.emphasis.label.offset .....	4083
6.19.708	chart.series.i[].leaves.emphasis.label.padding .....	4083
6.19.709	chart.series.i[].leaves.emphasis.label.position .....	4084
	Erlaubte Werte .....	4084
6.19.710	chart.series.i[].leaves.emphasis.label.rich.object.....	4085
6.19.711	chart.series.i[].leaves.emphasis.label.rotate .....	4086
6.19.712	chart.series.i[].leaves.emphasis.label.shadowBlur .....	4087
6.19.713	chart.series.i[].leaves.emphasis.label.shadowColor .....	4087
6.19.714	chart.series.i[].leaves.emphasis.label.shadowOffsetX .....	4087
6.19.715	chart.series.i[].leaves.emphasis.label.shadowOffsetY .....	4088
6.19.716	chart.series.i[].leaves.emphasis.label.show.....	4088
6.19.717	chart.series.i[].leaves.emphasis.label.textBorderColor .....	4088
6.19.718	chart.series.i[].leaves.emphasis.label.textBorderWidth.....	4089
6.19.719	chart.series.i[].leaves.emphasis.label.textShadowBlur .....	4089
6.19.720	chart.series.i[].leaves.emphasis.label.textShadowColor .....	4089
6.19.721	chart.series.i[].leaves.emphasis.label.textShadowOffsetX .....	4090
6.19.722	chart.series.i[].leaves.emphasis.label.textShadowOffsetY .....	4090
6.19.723	chart.series.i[].leaves.emphasis.label.verticalAlign.....	4091
	Erlaubte Werte .....	4091
6.19.724	chart.series.i[].leaves.emphasis.label.width .....	4091
6.19.725	chart.series.i[].leaves.emphasis.object .....	4092
6.19.726	chart.series.i[].leaves.itemStyle.borderColor .....	4092
6.19.727	chart.series.i[].leaves.itemStyle.borderType .....	4092
	Erlaubte Werte .....	4093
6.19.728	chart.series.i[].leaves.itemStyle.borderWidth.....	4093
6.19.729	chart.series.i[].leaves.itemStyle.color .....	4093
6.19.730	chart.series.i[].leaves.itemStyle.object .....	4095
6.19.731	chart.series.i[].leaves.itemStyle.opacity .....	4095
6.19.732	chart.series.i[].leaves.itemStyle.shadowBlur.....	4095
6.19.733	chart.series.i[].leaves.itemStyle.shadowColor.....	4096
6.19.734	chart.series.i[].leaves.itemStyle.shadowOffsetX.....	4096
6.19.735	chart.series.i[].leaves.itemStyle.shadowOffsetY.....	4096
6.19.736	chart.series.i[].leaves.label.align .....	4097
	Erlaubte Werte .....	4097
6.19.737	chart.series.i[].leaves.label.backgroundColor .....	4097

6.19.738	chart.series.i[].leaves.label.borderColor .....	4098
6.19.739	chart.series.i[].leaves.label.borderRadius .....	4099
6.19.740	chart.series.i[].leaves.label.borderWidth .....	4099
6.19.741	chart.series.i[].leaves.label.color .....	4099
6.19.742	chart.series.i[].leaves.label.distance .....	4100
6.19.743	chart.series.i[].leaves.label.fontFamily.....	4100
6.19.744	chart.series.i[].leaves.label.fontSize .....	4100
6.19.745	chart.series.i[].leaves.label.fontStyle .....	4101
	Erlaubte Werte .....	4101
6.19.746	chart.series.i[].leaves.label.fontWeight .....	4101
	Erlaubte Werte .....	4102
6.19.747	chart.series.i[].leaves.label.height .....	4102
6.19.748	chart.series.i[].leaves.label.lineHeight .....	4102
6.19.749	chart.series.i[].leaves.label.object .....	4103
6.19.750	chart.series.i[].leaves.label.offset .....	4103
6.19.751	chart.series.i[].leaves.label.padding.....	4103
6.19.752	chart.series.i[].leaves.label.position .....	4104
	Erlaubte Werte .....	4104
6.19.753	chart.series.i[].leaves.label.rich.object.....	4105
6.19.754	chart.series.i[].leaves.label.rotate .....	4106
6.19.755	chart.series.i[].leaves.label.shadowBlur.....	4107
6.19.756	chart.series.i[].leaves.label.shadowColor .....	4107
6.19.757	chart.series.i[].leaves.label.shadowOffsetX .....	4107
6.19.758	chart.series.i[].leaves.label.shadowOffsetY.....	4108
6.19.759	chart.series.i[].leaves.label.show.....	4108
6.19.760	chart.series.i[].leaves.label.textBorderColor.....	4108
6.19.761	chart.series.i[].leaves.label.textBorderWidth.....	4109
6.19.762	chart.series.i[].leaves.label.textShadowBlur.....	4109
6.19.763	chart.series.i[].leaves.label.textShadowColor .....	4109
6.19.764	chart.series.i[].leaves.label.textShadowOffsetX .....	4110
6.19.765	chart.series.i[].leaves.label.textShadowOffsetY.....	4110
6.19.766	chart.series.i[].leaves.label.verticalAlign.....	4111
	Erlaubte Werte .....	4111
6.19.767	chart.series.i[].leaves.label.width .....	4111
6.19.768	chart.series.i[].leaves.object .....	4112
6.19.769	chart.series.i[].left .....	4112

Erlaubte Werte .....	4113
6.19.770 chart.series.i[].legendHoverLink .....	4113
6.19.771 chart.series.i[].levels.childrenVisibleMin .....	4113
6.19.772 chart.series.i[].levels.colorAlpha .....	4114
6.19.773 chart.series.i[].levels.colorMappingBy .....	4114
Erlaubte Werte .....	4115
6.19.774 chart.series.i[].levels.colorSaturation .....	4115
6.19.775 chart.series.i[].levels.emphasis.itemStyle.color .....	4116
6.19.776 chart.series.i[].levels.emphasis.itemStyle.object .....	4116
6.19.777 chart.series.i[].levels.emphasis.label.align .....	4116
Erlaubte Werte .....	4117
6.19.778 chart.series.i[].levels.emphasis.label.backgroundColor .....	4117
6.19.779 chart.series.i[].levels.emphasis.label.borderColor .....	4118
6.19.780 chart.series.i[].levels.emphasis.label.borderRadius .....	4118
6.19.781 chart.series.i[].levels.emphasis.label.borderWidth .....	4119
6.19.782 chart.series.i[].levels.emphasis.label.color .....	4119
6.19.783 chart.series.i[].levels.emphasis.label.distance .....	4119
6.19.784 chart.series.i[].levels.emphasis.label.ellipsis .....	4120
6.19.785 chart.series.i[].levels.emphasis.label.fontFamily .....	4120
6.19.786 chart.series.i[].levels.emphasis.label.fontSize .....	4120
6.19.787 chart.series.i[].levels.emphasis.label.fontStyle .....	4121
Erlaubte Werte .....	4121
6.19.788 chart.series.i[].levels.emphasis.label.fontWeight .....	4121
Erlaubte Werte .....	4122
6.19.789 chart.series.i[].levels.emphasis.label.formatter .....	4122
6.19.790 chart.series.i[].levels.emphasis.label.height .....	4123
6.19.791 chart.series.i[].levels.emphasis.label.lineHeight .....	4124
6.19.792 chart.series.i[].levels.emphasis.label.object .....	4124
6.19.793 chart.series.i[].levels.emphasis.label.offset .....	4124
6.19.794 chart.series.i[].levels.emphasis.label.padding .....	4125
6.19.795 chart.series.i[].levels.emphasis.label.position .....	4125
Erlaubte Werte .....	4126
6.19.796 chart.series.i[].levels.emphasis.label.rich.object .....	4126
6.19.797 chart.series.i[].levels.emphasis.label.rotate .....	4127
6.19.798 chart.series.i[].levels.emphasis.label.shadowBlur .....	4128
6.19.799 chart.series.i[].levels.emphasis.label.shadowColor .....	4128

6.19.800	chart.series.i[].levels.emphasis.label.shadowOffsetX .....	4128
6.19.801	chart.series.i[].levels.emphasis.label.shadowOffsetY .....	4129
6.19.802	chart.series.i[].levels.emphasis.label.show.....	4129
6.19.803	chart.series.i[].levels.emphasis.label.textBorderColor .....	4129
6.19.804	chart.series.i[].levels.emphasis.label.textBorderWidth.....	4130
6.19.805	chart.series.i[].levels.emphasis.label.textShadowBlur .....	4130
6.19.806	chart.series.i[].levels.emphasis.label.textShadowColor .....	4130
6.19.807	chart.series.i[].levels.emphasis.label.textShadowOffsetX .....	4131
6.19.808	chart.series.i[].levels.emphasis.label.textShadowOffsetY .....	4131
6.19.809	chart.series.i[].levels.emphasis.label.verticalAlign.....	4132
	Erlaubte Werte .....	4132
6.19.810	chart.series.i[].levels.emphasis.label.width .....	4132
6.19.811	chart.series.i[].levels.emphasis.object .....	4133
6.19.812	chart.series.i[].levels.emphasis.upperLabel.align .....	4133
	Erlaubte Werte .....	4134
6.19.813	chart.series.i[].levels.emphasis.upperLabel.backgroundColor .....	4134
6.19.814	chart.series.i[].levels.emphasis.upperLabel.borderColor .....	4135
6.19.815	chart.series.i[].levels.emphasis.upperLabel.borderRadius.....	4135
6.19.816	chart.series.i[].levels.emphasis.upperLabel.borderWidth .....	4135
6.19.817	chart.series.i[].levels.emphasis.upperLabel.color .....	4136
6.19.818	chart.series.i[].levels.emphasis.upperLabel.distance .....	4136
6.19.819	chart.series.i[].levels.emphasis.upperLabel.ellipsis .....	4137
6.19.820	chart.series.i[].levels.emphasis.upperLabel.fontFamily .....	4137
6.19.821	chart.series.i[].levels.emphasis.upperLabel.fontSize.....	4137
6.19.822	chart.series.i[].levels.emphasis.upperLabel.fontStyle .....	4138
	Erlaubte Werte .....	4138
6.19.823	chart.series.i[].levels.emphasis.upperLabel.fontWeight.....	4138
	Erlaubte Werte .....	4139
6.19.824	chart.series.i[].levels.emphasis.upperLabel.formatter .....	4139
6.19.825	chart.series.i[].levels.emphasis.upperLabel.height.....	4140
6.19.826	chart.series.i[].levels.emphasis.upperLabel.lineHeight .....	4140
6.19.827	chart.series.i[].levels.emphasis.upperLabel.object.....	4141
6.19.828	chart.series.i[].levels.emphasis.upperLabel.offset .....	4141
6.19.829	chart.series.i[].levels.emphasis.upperLabel.padding.....	4142
6.19.830	chart.series.i[].levels.emphasis.upperLabel.position.....	4142
	Erlaubte Werte .....	4143



6.19.831	chart.series.i[].levels.emphasis.upperLabel.rich.object.....	4143
6.19.832	chart.series.i[].levels.emphasis.upperLabel.rotate .....	4144
6.19.833	chart.series.i[].levels.emphasis.upperLabel.shadowBlur .....	4145
6.19.834	chart.series.i[].levels.emphasis.upperLabel.shadowColor .....	4145
6.19.835	chart.series.i[].levels.emphasis.upperLabel.shadowOffsetX .....	4145
6.19.836	chart.series.i[].levels.emphasis.upperLabel.shadowOffsetY .....	4146
6.19.837	chart.series.i[].levels.emphasis.upperLabel.show.....	4146
6.19.838	chart.series.i[].levels.emphasis.upperLabel.textBorderColor.....	4146
6.19.839	chart.series.i[].levels.emphasis.upperLabel.textBorderWidth.....	4147
6.19.840	chart.series.i[].levels.emphasis.upperLabel.textShadowBlur.....	4147
6.19.841	chart.series.i[].levels.emphasis.upperLabel.textShadowColor .....	4147
6.19.842	chart.series.i[].levels.emphasis.upperLabel.textShadowOffsetX .....	4148
6.19.843	chart.series.i[].levels.emphasis.upperLabel.textShadowOffsetY.....	4148
6.19.844	chart.series.i[].levels.emphasis.upperLabel.verticalAlign.....	4149
	Erlaubte Werte .....	4149
6.19.845	chart.series.i[].levels.emphasis.upperLabel.width .....	4149
6.19.846	chart.series.i[].levels.itemStyle.borderColor .....	4150
6.19.847	chart.series.i[].levels.itemStyle.borderColorSaturation.....	4150
6.19.848	chart.series.i[].levels.itemStyle.borderWidth.....	4151
6.19.849	chart.series.i[].levels.itemStyle.color .....	4151
6.19.850	chart.series.i[].levels.itemStyle.colorAlpha.....	4151
6.19.851	chart.series.i[].levels.itemStyle.colorSaturation .....	4152
6.19.852	chart.series.i[].levels.itemStyle.gapWidth.....	4152
6.19.853	chart.series.i[].levels.itemStyle.object .....	4153
6.19.854	chart.series.i[].levels.itemStyle.strokeColor .....	4153
6.19.855	chart.series.i[].levels.itemStyle.strokeWidth .....	4153
6.19.856	chart.series.i[].levels.label.align .....	4154
	Erlaubte Werte .....	4154
6.19.857	chart.series.i[].levels.label.backgroundColor .....	4154
6.19.858	chart.series.i[].levels.label.borderColor .....	4155
6.19.859	chart.series.i[].levels.label.borderRadius.....	4155
6.19.860	chart.series.i[].levels.label.borderWidth .....	4156
6.19.861	chart.series.i[].levels.label.color .....	4156
6.19.862	chart.series.i[].levels.label.distance .....	4157
6.19.863	chart.series.i[].levels.label.ellipsis .....	4157
6.19.864	chart.series.i[].levels.label.fontFamily.....	4157

6.19.865	chart.series.i[].levels.label.fontSize .....	4158
6.19.866	chart.series.i[].levels.label.fontStyle .....	4158
6.19.867	chart.series.i[].levels.label.fontWeight .....	4158
	Erlaubte Werte .....	4159
6.19.868	chart.series.i[].levels.label.formatter .....	4159
6.19.869	chart.series.i[].levels.label.height .....	4160
6.19.870	chart.series.i[].levels.label.lineHeight .....	4161
6.19.871	chart.series.i[].levels.label.object .....	4161
6.19.872	chart.series.i[].levels.label.offset .....	4162
6.19.873	chart.series.i[].levels.label.padding.....	4162
6.19.874	chart.series.i[].levels.label.position .....	4162
	Erlaubte Werte .....	4163
6.19.875	chart.series.i[].levels.label.rich.object.....	4163
6.19.876	chart.series.i[].levels.label.rotate .....	4164
6.19.877	chart.series.i[].levels.label.shadowBlur.....	4165
6.19.878	chart.series.i[].levels.label.shadowColor .....	4165
6.19.879	chart.series.i[].levels.label.shadowOffsetX .....	4165
6.19.880	chart.series.i[].levels.label.shadowOffsetY.....	4166
6.19.881	chart.series.i[].levels.label.show.....	4166
6.19.882	chart.series.i[].levels.label.textBorderColor.....	4166
6.19.883	chart.series.i[].levels.label.textBorderWidth.....	4167
6.19.884	chart.series.i[].levels.label.textShadowBlur.....	4167
6.19.885	chart.series.i[].levels.label.textShadowColor .....	4167
6.19.886	chart.series.i[].levels.label.textShadowOffsetX .....	4168
6.19.887	chart.series.i[].levels.label.textShadowOffsetY.....	4168
6.19.888	chart.series.i[].levels.label.verticalAlign .....	4169
	Erlaubte Werte .....	4169
6.19.889	chart.series.i[].levels.label.width .....	4169
6.19.890	chart.series.i[].levels.object .....	4170
6.19.891	chart.series.i[].levels.upperLabel.align .....	4172
	Erlaubte Werte .....	4173
6.19.892	chart.series.i[].levels.upperLabel.backgroundColor .....	4173
6.19.893	chart.series.i[].levels.upperLabel.borderColor .....	4174
6.19.894	chart.series.i[].levels.upperLabel.borderRadius .....	4174
6.19.895	chart.series.i[].levels.upperLabel.borderWidth .....	4174
6.19.896	chart.series.i[].levels.upperLabel.color .....	4175

6.19.897	chart.series.i[].levels.upperLabel.distance .....	4175
6.19.898	chart.series.i[].levels.upperLabel.ellipsis .....	4176
6.19.899	chart.series.i[].levels.upperLabel.fontFamily.....	4176
6.19.900	chart.series.i[].levels.upperLabel.fontSize .....	4176
6.19.901	chart.series.i[].levels.upperLabel.fontStyle .....	4177
	Erlaubte Werte .....	4177
6.19.902	chart.series.i[].levels.upperLabel.fontWeight .....	4177
	Erlaubte Werte .....	4178
6.19.903	chart.series.i[].levels.upperLabel.formatter.....	4178
6.19.904	chart.series.i[].levels.upperLabel.height .....	4179
6.19.905	chart.series.i[].levels.upperLabel.lineHeight .....	4179
6.19.906	chart.series.i[].levels.upperLabel.object .....	4180
6.19.907	chart.series.i[].levels.upperLabel.offset .....	4180
6.19.908	chart.series.i[].levels.upperLabel.padding.....	4181
6.19.909	chart.series.i[].levels.upperLabel.position .....	4181
	Erlaubte Werte .....	4182
6.19.910	chart.series.i[].levels.upperLabel.rich.object.....	4182
6.19.911	chart.series.i[].levels.upperLabel.rotate .....	4183
6.19.912	chart.series.i[].levels.upperLabel.shadowBlur.....	4184
6.19.913	chart.series.i[].levels.upperLabel.shadowColor .....	4184
6.19.914	chart.series.i[].levels.upperLabel.shadowOffsetX .....	4184
6.19.915	chart.series.i[].levels.upperLabel.shadowOffsetY.....	4185
6.19.916	chart.series.i[].levels.upperLabel.show .....	4185
6.19.917	chart.series.i[].levels.upperLabel.textBorderColor.....	4185
6.19.918	chart.series.i[].levels.upperLabel.textBorderWidth.....	4186
6.19.919	chart.series.i[].levels.upperLabel.textShadowBlur.....	4186
6.19.920	chart.series.i[].levels.upperLabel.textShadowColor.....	4186
6.19.921	chart.series.i[].levels.upperLabel.textShadowOffsetX .....	4187
6.19.922	chart.series.i[].levels.upperLabel.textShadowOffsetY.....	4187
6.19.923	chart.series.i[].levels.upperLabel.verticalAlign .....	4188
	Erlaubte Werte .....	4188
6.19.924	chart.series.i[].levels.upperLabel.width .....	4188
6.19.925	chart.series.i[].levels.visibleMin .....	4189
6.19.926	chart.series.i[].levels.visualDimension .....	4189
6.19.927	chart.series.i[].levels.visualMax .....	4190
6.19.928	chart.series.i[].levels.visualMin .....	4190

6.19.929	chart.series.i[].lineStyle.color .....	4191
6.19.930	chart.series.i[].lineStyle.object .....	4192
6.19.931	chart.series.i[].lineStyle.opacity .....	4193
6.19.932	chart.series.i[].lineStyle.shadowBlur .....	4193
6.19.933	chart.series.i[].lineStyle.shadowColor .....	4194
6.19.934	chart.series.i[].lineStyle.shadowOffsetX .....	4194
6.19.935	chart.series.i[].lineStyle.shadowOffsetY .....	4194
6.19.936	chart.series.i[].lineStyle.type .....	4195
	Erlaubte Werte .....	4195
6.19.937	chart.series.i[].lineStyle.width .....	4195
6.19.938	chart.series.i[].links.emphasis.label.align .....	4195
	Erlaubte Werte .....	4196
6.19.939	chart.series.i[].links.emphasis.label.backgroundColor .....	4196
6.19.940	chart.series.i[].links.emphasis.label.borderColor .....	4197
6.19.941	chart.series.i[].links.emphasis.label.borderRadius .....	4197
6.19.942	chart.series.i[].links.emphasis.label.borderWidth .....	4198
6.19.943	chart.series.i[].links.emphasis.label.color .....	4198
6.19.944	chart.series.i[].links.emphasis.label.fontFamily .....	4198
6.19.945	chart.series.i[].links.emphasis.label.fontSize .....	4199
6.19.946	chart.series.i[].links.emphasis.label.fontStyle .....	4199
	Erlaubte Werte .....	4199
6.19.947	chart.series.i[].links.emphasis.label.fontWeight .....	4199
	Erlaubte Werte .....	4200
6.19.948	chart.series.i[].links.emphasis.label.formatter .....	4200
6.19.949	chart.series.i[].links.emphasis.label.height .....	4201
6.19.950	chart.series.i[].links.emphasis.label.lineHeight .....	4202
6.19.951	chart.series.i[].links.emphasis.label.object .....	4202
6.19.952	chart.series.i[].links.emphasis.label.padding .....	4203
6.19.953	chart.series.i[].links.emphasis.label.position .....	4203
	Erlaubte Werte .....	4204
6.19.954	chart.series.i[].links.emphasis.label.rich.object .....	4204
6.19.955	chart.series.i[].links.emphasis.label.shadowBlur .....	4205
6.19.956	chart.series.i[].links.emphasis.label.shadowColor .....	4206
6.19.957	chart.series.i[].links.emphasis.label.shadowOffsetX .....	4206
6.19.958	chart.series.i[].links.emphasis.label.shadowOffsetY .....	4206
6.19.959	chart.series.i[].links.emphasis.label.show .....	4207

6.19.960	chart.series.i[].links.emphasis.label.textBorderColor .....	4207
6.19.961	chart.series.i[].links.emphasis.label.textBorderWidth .....	4207
6.19.962	chart.series.i[].links.emphasis.label.textShadowBlur .....	4208
6.19.963	chart.series.i[].links.emphasis.label.textShadowColor .....	4208
6.19.964	chart.series.i[].links.emphasis.label.textShadowOffsetX .....	4208
6.19.965	chart.series.i[].links.emphasis.label.textShadowOffsetY .....	4209
6.19.966	chart.series.i[].links.emphasis.label.verticalAlign .....	4209
	Erlaubte Werte .....	4210
6.19.967	chart.series.i[].links.emphasis.label.width.....	4210
6.19.968	chart.series.i[].links.emphasis.lineStyle.color .....	4210
6.19.969	chart.series.i[].links.emphasis.lineStyle.object .....	4212
6.19.970	chart.series.i[].links.emphasis.lineStyle.opacity .....	4212
6.19.971	chart.series.i[].links.emphasis.lineStyle.shadowBlur .....	4212
6.19.972	chart.series.i[].links.emphasis.lineStyle.shadowColor .....	4213
6.19.973	chart.series.i[].links.emphasis.lineStyle.shadowOffsetX.....	4213
6.19.974	chart.series.i[].links.emphasis.lineStyle.shadowOffsetY.....	4213
6.19.975	chart.series.i[].links.emphasis.lineStyle.type.....	4214
	Erlaubte Werte .....	4214
6.19.976	chart.series.i[].links.emphasis.lineStyle.width .....	4214
6.19.977	chart.series.i[].links.emphasis.object .....	4215
6.19.978	chart.series.i[].links.label.align .....	4215
	Erlaubte Werte .....	4215
6.19.979	chart.series.i[].links.label.backgroundColor .....	4216
6.19.980	chart.series.i[].links.label.borderColor .....	4216
6.19.981	chart.series.i[].links.label.borderRadius.....	4217
6.19.982	chart.series.i[].links.label.borderWidth .....	4217
6.19.983	chart.series.i[].links.label.color .....	4217
6.19.984	chart.series.i[].links.label.fontFamily .....	4218
6.19.985	chart.series.i[].links.label.fontSize.....	4218
6.19.986	chart.series.i[].links.label.fontStyle .....	4218
	Erlaubte Werte .....	4219
6.19.987	chart.series.i[].links.label.fontWeight.....	4219
	Erlaubte Werte .....	4219
6.19.988	chart.series.i[].links.label.formatter .....	4220
6.19.989	chart.series.i[].links.label.height.....	4221
6.19.990	chart.series.i[].links.label.lineHeight.....	4221

6.19.991	chart.series.i[].links.label.object.....	4222
6.19.992	chart.series.i[].links.label.padding .....	4222
6.19.993	chart.series.i[].links.label.position.....	4223
	Erlaubte Werte .....	4223
6.19.994	chart.series.i[].links.label.rich.object .....	4224
6.19.995	chart.series.i[].links.label.shadowBlur .....	4225
6.19.996	chart.series.i[].links.label.shadowColor .....	4226
6.19.997	chart.series.i[].links.label.shadowOffsetX .....	4226
6.19.998	chart.series.i[].links.label.shadowOffsetY .....	4226
6.19.999	chart.series.i[].links.label.show .....	4227
6.19.1000	chart.series.i[].links.label.textBorderColor .....	4227
6.19.1001	chart.series.i[].links.label.textBorderWidth .....	4227
6.19.1002	chart.series.i[].links.label.textShadowBlur .....	4228
6.19.1003	chart.series.i[].links.label.textShadowColor .....	4228
6.19.1004	chart.series.i[].links.label.textShadowOffsetX .....	4228
6.19.1005	chart.series.i[].links.label.textShadowOffsetY .....	4229
6.19.1006	chart.series.i[].links.label.verticalAlign .....	4229
	Erlaubte Werte .....	4230
6.19.1007	chart.series.i[].links.label.width.....	4230
6.19.1008	chart.series.i[].links.lineStyle.color .....	4230
6.19.1009	chart.series.i[].links.lineStyle.curveness .....	4232
6.19.1010	chart.series.i[].links.lineStyle.object.....	4232
6.19.1011	chart.series.i[].links.lineStyle.opacity.....	4232
6.19.1012	chart.series.i[].links.lineStyle.shadowBlur .....	4233
6.19.1013	chart.series.i[].links.lineStyle.shadowColor.....	4233
6.19.1014	chart.series.i[].links.lineStyle.shadowOffsetX.....	4233
6.19.1015	chart.series.i[].links.lineStyle.shadowOffsetY.....	4234
6.19.1016	chart.series.i[].links.lineStyle.type.....	4234
	Erlaubte Werte .....	4234
6.19.1017	chart.series.i[].links.lineStyle.width .....	4235
6.19.1018	chart.series.i[].links.object .....	4235
6.19.1019	chart.series.i[].links.source .....	4235
6.19.1020	chart.series.i[].links.symbol .....	4236
6.19.1021	chart.series.i[].links.symbolSize.....	4236
6.19.1022	chart.series.i[].links.target.....	4236
6.19.1023	chart.series.i[].links.value.....	4237

6.19.1024	chart.series.i[].markArea.animation .....	4237
6.19.1025	chart.series.i[].markArea.animationDelay.....	4238
6.19.1026	chart.series.i[].markArea.animationDelayUpdate .....	4238
6.19.1027	chart.series.i[].markArea.animationDuration .....	4239
6.19.1028	chart.series.i[].markArea.animationDurationUpdate .....	4239
6.19.1029	chart.series.i[].markArea.animationEasing.....	4240
	Erlaubte Werte .....	4240
6.19.1030	chart.series.i[].markArea.animationEasingUpdate .....	4241
	Erlaubte Werte .....	4241
6.19.1031	chart.series.i[].markArea.animationThreshold .....	4242
6.19.1032	chart.series.i[].markArea.data.object .....	4242
6.19.1033	chart.series.i[].markArea.itemStyle.borderColor.....	4244
6.19.1034	chart.series.i[].markArea.itemStyle.borderType.....	4244
	Erlaubte Werte .....	4244
6.19.1035	chart.series.i[].markArea.itemStyle.borderWidth .....	4244
6.19.1036	chart.series.i[].markArea.itemStyle.color.....	4245
6.19.1037	chart.series.i[].markArea.itemStyle.emphasis.borderColor.....	4246
6.19.1038	chart.series.i[].markArea.itemStyle.emphasis.borderType .....	4247
	Erlaubte Werte .....	4247
6.19.1039	chart.series.i[].markArea.itemStyle.emphasis.borderWidth.....	4247
6.19.1040	chart.series.i[].markArea.itemStyle.emphasis.color .....	4248
6.19.1041	chart.series.i[].markArea.itemStyle.emphasis.object .....	4249
6.19.1042	chart.series.i[].markArea.itemStyle.emphasis.opacity.....	4250
6.19.1043	chart.series.i[].markArea.itemStyle.emphasis.shadowBlur .....	4250
6.19.1044	chart.series.i[].markArea.itemStyle.emphasis.shadowColor .....	4251
6.19.1045	chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetX.....	4251
6.19.1046	chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetY .....	4251
6.19.1047	chart.series.i[].markArea.itemStyle.object.....	4252
6.19.1048	chart.series.i[].markArea.itemStyle.opacity.....	4252
6.19.1049	chart.series.i[].markArea.itemStyle.shadowBlur .....	4252
6.19.1050	chart.series.i[].markArea.itemStyle.shadowColor .....	4253
6.19.1051	chart.series.i[].markArea.itemStyle.shadowOffsetX.....	4253
6.19.1052	chart.series.i[].markArea.itemStyle.shadowOffsetY .....	4253
6.19.1053	chart.series.i[].markArea.label.align.....	4254
	Erlaubte Werte .....	4254
6.19.1054	chart.series.i[].markArea.label.backgroundColor.....	4255

6.19.1055	chart.series.i[].markArea.label.borderColor .....	4255
6.19.1056	chart.series.i[].markArea.label.borderRadius .....	4256
6.19.1057	chart.series.i[].markArea.label.borderWidth.....	4256
6.19.1058	chart.series.i[].markArea.label.color .....	4256
6.19.1059	chart.series.i[].markArea.label.distance.....	4257
6.19.1060	chart.series.i[].markArea.label.emphasis.align.....	4257
	Erlaubte Werte .....	4258
6.19.1061	chart.series.i[].markArea.label.emphasis.backgroundColor .....	4258
6.19.1062	chart.series.i[].markArea.label.emphasis.borderColor .....	4259
6.19.1063	chart.series.i[].markArea.label.emphasis.borderRadius .....	4259
6.19.1064	chart.series.i[].markArea.label.emphasis.borderWidth.....	4259
6.19.1065	chart.series.i[].markArea.label.emphasis.color .....	4260
6.19.1066	chart.series.i[].markArea.label.emphasis.distance.....	4260
6.19.1067	chart.series.i[].markArea.label.emphasis.fontFamily.....	4260
6.19.1068	chart.series.i[].markArea.label.emphasis.fontSize .....	4261
6.19.1069	chart.series.i[].markArea.label.emphasis.fontStyle.....	4261
	Erlaubte Werte .....	4262
6.19.1070	chart.series.i[].markArea.label.emphasis.fontWeight .....	4262
	Erlaubte Werte .....	4262
6.19.1071	chart.series.i[].markArea.label.emphasis.height .....	4262
6.19.1072	chart.series.i[].markArea.label.emphasis.lineHeight .....	4263
6.19.1073	chart.series.i[].markArea.label.emphasis.object .....	4263
6.19.1074	chart.series.i[].markArea.label.emphasis.offset .....	4264
6.19.1075	chart.series.i[].markArea.label.emphasis.padding.....	4264
6.19.1076	chart.series.i[].markArea.label.emphasis.position .....	4264
	Erlaubte Werte .....	4265
6.19.1077	chart.series.i[].markArea.label.emphasis.rich.object.....	4265
6.19.1078	chart.series.i[].markArea.label.emphasis.rotate.....	4266
6.19.1079	chart.series.i[].markArea.label.emphasis.shadowBlur.....	4267
6.19.1080	chart.series.i[].markArea.label.emphasis.shadowColor.....	4267
6.19.1081	chart.series.i[].markArea.label.emphasis.shadowOffsetX.....	4267
6.19.1082	chart.series.i[].markArea.label.emphasis.shadowOffsetY.....	4268
6.19.1083	chart.series.i[].markArea.label.emphasis.show .....	4268
6.19.1084	chart.series.i[].markArea.label.emphasis.textBorderColor.....	4268
6.19.1085	chart.series.i[].markArea.label.emphasis.textBorderWidth.....	4269
6.19.1086	chart.series.i[].markArea.label.emphasis.textShadowBlur.....	4269



6.19.1087	chart.series.i[].markArea.label.emphasis.textShadowColor.....	4270
6.19.1088	chart.series.i[].markArea.label.emphasis.textShadowOffsetX.....	4270
6.19.1089	chart.series.i[].markArea.label.emphasis.textShadowOffsetY.....	4270
6.19.1090	chart.series.i[].markArea.label.emphasis.verticalAlign .....	4271
	Erlaubte Werte .....	4271
6.19.1091	chart.series.i[].markArea.label.emphasis.width .....	4271
6.19.1092	chart.series.i[].markArea.label.fontFamily .....	4272
6.19.1093	chart.series.i[].markArea.label.fontSize .....	4272
6.19.1094	chart.series.i[].markArea.label.fontStyle.....	4273
	Erlaubte Werte .....	4273
6.19.1095	chart.series.i[].markArea.label.fontWeight .....	4273
	Erlaubte Werte .....	4273
6.19.1096	chart.series.i[].markArea.label.height .....	4274
6.19.1097	chart.series.i[].markArea.label.lineHeight.....	4274
6.19.1098	chart.series.i[].markArea.label.object .....	4275
6.19.1099	chart.series.i[].markArea.label.offset .....	4275
6.19.1100	chart.series.i[].markArea.label.padding .....	4275
6.19.1101	chart.series.i[].markArea.label.position .....	4276
	Erlaubte Werte .....	4276
6.19.1102	chart.series.i[].markArea.label.rich.object.....	4277
6.19.1103	chart.series.i[].markArea.label.rotate.....	4278
6.19.1104	chart.series.i[].markArea.label.shadowBlur.....	4279
6.19.1105	chart.series.i[].markArea.label.shadowColor.....	4279
6.19.1106	chart.series.i[].markArea.label.shadowOffsetX.....	4279
6.19.1107	chart.series.i[].markArea.label.shadowOffsetY .....	4280
6.19.1108	chart.series.i[].markArea.label.show .....	4280
6.19.1109	chart.series.i[].markArea.label.textBorderColor.....	4280
6.19.1110	chart.series.i[].markArea.label.textBorderWidth .....	4281
6.19.1111	chart.series.i[].markArea.label.textShadowBlur.....	4281
6.19.1112	chart.series.i[].markArea.label.textShadowColor.....	4281
6.19.1113	chart.series.i[].markArea.label.textShadowOffsetX.....	4282
6.19.1114	chart.series.i[].markArea.label.textShadowOffsetY .....	4282
6.19.1115	chart.series.i[].markArea.label.verticalAlign .....	4283
	Erlaubte Werte .....	4283
6.19.1116	chart.series.i[].markArea.label.width .....	4283
6.19.1117	chart.series.i[].markArea.object.....	4284

6.19.1118	chart.series.i[].markArea.silent .....	4284
6.19.1119	chart.series.i[].markLine.animation .....	4284
6.19.1120	chart.series.i[].markLine.animationDelay .....	4285
6.19.1121	chart.series.i[].markLine.animationDelayUpdate .....	4286
6.19.1122	chart.series.i[].markLine.animationDuration .....	4286
6.19.1123	chart.series.i[].markLine.animationDurationUpdate .....	4287
6.19.1124	chart.series.i[].markLine.animationEasing .....	4287
	Erlaubte Werte .....	4287
6.19.1125	chart.series.i[].markLine.animationEasingUpdate .....	4288
	Erlaubte Werte .....	4288
6.19.1126	chart.series.i[].markLine.animationThreshold .....	4289
6.19.1127	chart.series.i[].markLine.label.emphasis.formatter .....	4289
6.19.1128	chart.series.i[].markLine.label.emphasis.object .....	4291
6.19.1129	chart.series.i[].markLine.label.emphasis.position .....	4291
	Erlaubte Werte .....	4291
6.19.1130	chart.series.i[].markLine.label.emphasis.show .....	4291
6.19.1131	chart.series.i[].markLine.label.formatter .....	4292
6.19.1132	chart.series.i[].markLine.label.object .....	4293
6.19.1133	chart.series.i[].markLine.label.position .....	4293
	Erlaubte Werte .....	4294
6.19.1134	chart.series.i[].markLine.label.show .....	4294
6.19.1135	chart.series.i[].markLine.lineStyle.color .....	4294
6.19.1136	chart.series.i[].markLine.lineStyle.curveness .....	4295
6.19.1137	chart.series.i[].markLine.lineStyle.emphasis.color .....	4296
6.19.1138	chart.series.i[].markLine.lineStyle.emphasis.object .....	4297
6.19.1139	chart.series.i[].markLine.lineStyle.emphasis.opacity .....	4298
6.19.1140	chart.series.i[].markLine.lineStyle.emphasis.shadowBlur .....	4298
6.19.1141	chart.series.i[].markLine.lineStyle.emphasis.shadowColor .....	4299
6.19.1142	chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetX .....	4299
6.19.1143	chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetY .....	4299
6.19.1144	chart.series.i[].markLine.lineStyle.emphasis.type .....	4300
	Erlaubte Werte .....	4300
6.19.1145	chart.series.i[].markLine.lineStyle.emphasis.width .....	4300
6.19.1146	chart.series.i[].markLine.lineStyle.object .....	4300
6.19.1147	chart.series.i[].markLine.lineStyle.opacity .....	4301
6.19.1148	chart.series.i[].markLine.lineStyle.shadowBlur .....	4301

6.19.1149	chart.series.i[].markLine.lineStyle.shadowColor .....	4302
6.19.1150	chart.series.i[].markLine.lineStyle.shadowOffsetX .....	4302
6.19.1151	chart.series.i[].markLine.lineStyle.shadowOffsetY .....	4302
6.19.1152	chart.series.i[].markLine.lineStyle.type .....	4303
	Erlaubte Werte .....	4303
6.19.1153	chart.series.i[].markLine.lineStyle.width .....	4303
6.19.1154	chart.series.i[].markLine.object .....	4304
6.19.1155	chart.series.i[].markLine.precision .....	4304
6.19.1156	chart.series.i[].markLine.silent .....	4304
6.19.1157	chart.series.i[].markLine.symbol .....	4305
6.19.1158	chart.series.i[].markLine.symbolSize .....	4305
6.19.1159	chart.series.i[].markPoint.animation .....	4305
6.19.1160	chart.series.i[].markPoint.animationDelay .....	4306
6.19.1161	chart.series.i[].markPoint.animationDelayUpdate .....	4306
6.19.1162	chart.series.i[].markPoint.animationDuration .....	4307
6.19.1163	chart.series.i[].markPoint.animationDurationUpdate .....	4307
6.19.1164	chart.series.i[].markPoint.animationEasing .....	4308
	Erlaubte Werte .....	4308
6.19.1165	chart.series.i[].markPoint.animationEasingUpdate .....	4309
	Erlaubte Werte .....	4309
6.19.1166	chart.series.i[].markPoint.animationThreshold .....	4310
6.19.1167	chart.series.i[].markPoint.itemStyle.borderColor .....	4310
6.19.1168	chart.series.i[].markPoint.itemStyle.borderType .....	4310
	Erlaubte Werte .....	4311
6.19.1169	chart.series.i[].markPoint.itemStyle.borderWidth .....	4311
6.19.1170	chart.series.i[].markPoint.itemStyle.color .....	4311
6.19.1171	chart.series.i[].markPoint.itemStyle.emphasis.borderColor .....	4313
6.19.1172	chart.series.i[].markPoint.itemStyle.emphasis.borderType .....	4313
	Erlaubte Werte .....	4313
6.19.1173	chart.series.i[].markPoint.itemStyle.emphasis.borderWidth .....	4313
6.19.1174	chart.series.i[].markPoint.itemStyle.emphasis.color .....	4314
6.19.1175	chart.series.i[].markPoint.itemStyle.emphasis.object .....	4315
6.19.1176	chart.series.i[].markPoint.itemStyle.emphasis.opacity .....	4316
6.19.1177	chart.series.i[].markPoint.itemStyle.emphasis.shadowBlur .....	4316
6.19.1178	chart.series.i[].markPoint.itemStyle.emphasis.shadowColor .....	4317
6.19.1179	chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetX .....	4317

6.19.1180	chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetY.....	4317
6.19.1181	chart.series.i[].markPoint.itemStyle.object .....	4318
6.19.1182	chart.series.i[].markPoint.itemStyle.opacity .....	4318
6.19.1183	chart.series.i[].markPoint.itemStyle.shadowBlur.....	4318
6.19.1184	chart.series.i[].markPoint.itemStyle.shadowColor.....	4319
6.19.1185	chart.series.i[].markPoint.itemStyle.shadowOffsetX .....	4319
6.19.1186	chart.series.i[].markPoint.itemStyle.shadowOffsetY.....	4319
6.19.1187	chart.series.i[].markPoint.label.align .....	4320
	Erlaubte Werte .....	4320
6.19.1188	chart.series.i[].markPoint.label.backgroundColor .....	4321
6.19.1189	chart.series.i[].markPoint.label.borderColor .....	4321
6.19.1190	chart.series.i[].markPoint.label.borderRadius.....	4322
6.19.1191	chart.series.i[].markPoint.label.borderWidth .....	4322
6.19.1192	chart.series.i[].markPoint.label.color .....	4322
6.19.1193	chart.series.i[].markPoint.label.distance .....	4323
6.19.1194	chart.series.i[].markPoint.label.emphasis.align .....	4323
	Erlaubte Werte .....	4324
6.19.1195	chart.series.i[].markPoint.label.emphasis.backgroundColor .....	4324
6.19.1196	chart.series.i[].markPoint.label.emphasis.borderColor .....	4325
6.19.1197	chart.series.i[].markPoint.label.emphasis.borderRadius.....	4325
6.19.1198	chart.series.i[].markPoint.label.emphasis.borderWidth .....	4325
6.19.1199	chart.series.i[].markPoint.label.emphasis.color .....	4326
6.19.1200	chart.series.i[].markPoint.label.emphasis.distance .....	4326
6.19.1201	chart.series.i[].markPoint.label.emphasis.fontFamily .....	4326
6.19.1202	chart.series.i[].markPoint.label.emphasis.fontSize.....	4327
6.19.1203	chart.series.i[].markPoint.label.emphasis.fontStyle .....	4327
	Erlaubte Werte .....	4328
6.19.1204	chart.series.i[].markPoint.label.emphasis.fontWeight.....	4328
	Erlaubte Werte .....	4328
6.19.1205	chart.series.i[].markPoint.label.emphasis.formatter .....	4328
6.19.1206	chart.series.i[].markPoint.label.emphasis.height.....	4329
6.19.1207	chart.series.i[].markPoint.label.emphasis.lineHeight .....	4330
6.19.1208	chart.series.i[].markPoint.label.emphasis.object.....	4330
6.19.1209	chart.series.i[].markPoint.label.emphasis.padding.....	4331
6.19.1210	chart.series.i[].markPoint.label.emphasis.position.....	4331
	Erlaubte Werte .....	4332

6.19.1211	chart.series.i[].markPoint.label.emphasis.rich.object .....	4332
6.19.1212	chart.series.i[].markPoint.label.emphasis.rotate .....	4333
6.19.1213	chart.series.i[].markPoint.label.emphasis.shadowBlur .....	4334
6.19.1214	chart.series.i[].markPoint.label.emphasis.shadowColor .....	4334
6.19.1215	chart.series.i[].markPoint.label.emphasis.shadowOffsetX .....	4334
6.19.1216	chart.series.i[].markPoint.label.emphasis.shadowOffsetY .....	4335
6.19.1217	chart.series.i[].markPoint.label.emphasis.show.....	4335
6.19.1218	chart.series.i[].markPoint.label.emphasis.textBorderColor .....	4335
6.19.1219	chart.series.i[].markPoint.label.emphasis.textBorderWidth .....	4336
6.19.1220	chart.series.i[].markPoint.label.emphasis.textShadowBlur .....	4336
6.19.1221	chart.series.i[].markPoint.label.emphasis.textShadowColor .....	4336
6.19.1222	chart.series.i[].markPoint.label.emphasis.textShadowOffsetX .....	4337
6.19.1223	chart.series.i[].markPoint.label.emphasis.textShadowOffsetY .....	4337
6.19.1224	chart.series.i[].markPoint.label.emphasis.verticalAlign.....	4338
	Erlaubte Werte .....	4338
6.19.1225	chart.series.i[].markPoint.label.emphasis.width .....	4338
6.19.1226	chart.series.i[].markPoint.label.fontFamily .....	4339
6.19.1227	chart.series.i[].markPoint.label.fontSize .....	4339
6.19.1228	chart.series.i[].markPoint.label.fontStyle .....	4340
	Erlaubte Werte .....	4340
6.19.1229	chart.series.i[].markPoint.label.fontWeight.....	4340
	Erlaubte Werte .....	4340
6.19.1230	chart.series.i[].markPoint.label.formatter .....	4341
6.19.1231	chart.series.i[].markPoint.label.height .....	4342
6.19.1232	chart.series.i[].markPoint.label.lineHeight .....	4342
6.19.1233	chart.series.i[].markPoint.label.object .....	4343
6.19.1234	chart.series.i[].markPoint.label.padding.....	4343
6.19.1235	chart.series.i[].markPoint.label.position .....	4344
	Erlaubte Werte .....	4344
6.19.1236	chart.series.i[].markPoint.label.rich.object.....	4345
6.19.1237	chart.series.i[].markPoint.label.rotate .....	4346
6.19.1238	chart.series.i[].markPoint.label.shadowBlur .....	4347
6.19.1239	chart.series.i[].markPoint.label.shadowColor .....	4347
6.19.1240	chart.series.i[].markPoint.label.shadowOffsetX .....	4347
6.19.1241	chart.series.i[].markPoint.label.shadowOffsetY .....	4348
6.19.1242	chart.series.i[].markPoint.label.show.....	4348

6.19.1243	chart.series.i[].markPoint.label.textBorderColor .....	4348
6.19.1244	chart.series.i[].markPoint.label.textBorderWidth.....	4349
6.19.1245	chart.series.i[].markPoint.label.textShadowBlur .....	4349
6.19.1246	chart.series.i[].markPoint.label.textShadowColor .....	4349
6.19.1247	chart.series.i[].markPoint.label.textShadowOffsetX .....	4350
6.19.1248	chart.series.i[].markPoint.label.textShadowOffsetY .....	4350
6.19.1249	chart.series.i[].markPoint.label.verticalAlign.....	4351
	Erlaubte Werte .....	4351
6.19.1250	chart.series.i[].markPoint.label.width .....	4351
6.19.1251	chart.series.i[].markPoint.object .....	4352
6.19.1252	chart.series.i[].markPoint.silent.....	4352
6.19.1253	chart.series.i[].markPoint.symbol.....	4352
	Erlaubte Werte .....	4353
6.19.1254	chart.series.i[].markPoint.symbolKeepAspect.....	4353
6.19.1255	chart.series.i[].markPoint.symbolOffset .....	4354
	Erlaubte Werte .....	4354
6.19.1256	chart.series.i[].markPoint.symbolRotate .....	4354
6.19.1257	chart.series.i[].markPoint.symbolSize.....	4354
6.19.1258	chart.series.i[].max .....	4355
6.19.1259	chart.series.i[].maxOpacity .....	4355
6.19.1260	chart.series.i[].maxSize.....	4356
6.19.1261	chart.series.i[].min .....	4356
6.19.1262	chart.series.i[].minAngle.....	4356
6.19.1263	chart.series.i[].minOpacity .....	4357
6.19.1264	chart.series.i[].minSize .....	4357
6.19.1265	chart.series.i[].nameMap.....	4358
6.19.1266	chart.series.i[].nodeClick.....	4358
6.19.1267	chart.series.i[].nodeGap .....	4359
6.19.1268	chart.series.i[].nodes .....	4359
6.19.1269	chart.series.i[].nodeScaleRatio .....	4359
6.19.1270	chart.series.i[].nodeWidth .....	4360
6.19.1271	chart.series.i[].orient .....	4360
	Erlaubte Werte .....	4361
6.19.1272	chart.series.i[].parallellIndex.....	4361
6.19.1273	chart.series.i[].pointer.length .....	4361
	Erlaubte Werte .....	4361

6.19.1274	chart.series.i[].pointer.object.....	4362
6.19.1275	chart.series.i[].pointer.show .....	4362
6.19.1276	chart.series.i[].pointer.width.....	4362
6.19.1277	chart.series.i[].polarIndex .....	4363
6.19.1278	chart.series.i[].polyline .....	4363
6.19.1279	chart.series.i[].progressive .....	4363
6.19.1280	chart.series.i[].progressiveChunkMode .....	4364
	Erlaubte Werte .....	4364
6.19.1281	chart.series.i[].progressiveThreshold .....	4364
6.19.1282	chart.series.i[].radarIndex .....	4365
6.19.1283	chart.series.i[].radius .....	4365
	Erlaubte Werte .....	4366
6.19.1284	chart.series.i[].realtime .....	4366
6.19.1285	chart.series.i[].renderItem.object .....	4366
6.19.1286	chart.series.i[].renderLabelForZeroData .....	4368
6.19.1287	chart.series.i[].right.....	4368
	Erlaubte Werte .....	4368
6.19.1288	chart.series.i[].roam.....	4368
	Erlaubte Werte .....	4369
6.19.1289	chart.series.i[].roseType .....	4369
	Erlaubte Werte .....	4369
6.19.1290	chart.series.i[].sampling .....	4369
	Erlaubte Werte .....	4370
6.19.1291	chart.series.i[].selectedMode .....	4370
	Erlaubte Werte .....	4370
6.19.1292	chart.series.i[].selectedOffset .....	4371
6.19.1293	chart.series.i[].seriesLayoutBy .....	4371
	Erlaubte Werte .....	4371
6.19.1294	chart.series.i[].showAllSymbol.....	4372
	Erlaubte Werte .....	4372
6.19.1295	chart.series.i[].showEffectOn .....	4372
	Erlaubte Werte .....	4373
6.19.1296	chart.series.i[].showSymbol.....	4373
6.19.1297	chart.series.i[].silent .....	4373
6.19.1298	chart.series.i[].silent.link .....	4373
6.19.1299	chart.series.i[].silent.object.....	4374

6.19.1300 chart.series.i[].silent.target .....	4374
Erlaubte Werte .....	4375
6.19.1301 chart.series.i[].silent.tooltip.backgroundColor.....	4375
6.19.1302 chart.series.i[].silent.tooltip.borderColor.....	4375
6.19.1303 chart.series.i[].silent.tooltip.borderWidth.....	4375
6.19.1304 chart.series.i[].silent.tooltip.extraCSSText.....	4376
6.19.1305 chart.series.i[].silent.tooltip.formatter .....	4376
6.19.1306 chart.series.i[].silent.tooltip.object .....	4379
6.19.1307 chart.series.i[].silent.tooltip.padding.....	4379
6.19.1308 chart.series.i[].silent.tooltip.position .....	4380
6.19.1309 chart.series.i[].silent.tooltip.textStyle.color.....	4381
6.19.1310 chart.series.i[].silent.tooltip.textStyle.fontFamily .....	4382
6.19.1311 chart.series.i[].silent.tooltip.textStyle.fontSize .....	4382
6.19.1312 chart.series.i[].silent.tooltip.textStyle.fontStyle.....	4383
Erlaubte Werte .....	4383
6.19.1313 chart.series.i[].silent.tooltip.textStyle.fontWeight.....	4383
Erlaubte Werte .....	4383
6.19.1314 chart.series.i[].silent.tooltip.textStyle.height .....	4384
6.19.1315 chart.series.i[].silent.tooltip.textStyle.lineHeight.....	4384
6.19.1316 chart.series.i[].silent.tooltip.textStyle.object.....	4385
6.19.1317 chart.series.i[].silent.tooltip.textStyle.textBorderColor .....	4385
6.19.1318 chart.series.i[].silent.tooltip.textStyle.textBorderWidth .....	4385
6.19.1319 chart.series.i[].silent.tooltip.textStyle.textShadowBlur .....	4386
6.19.1320 chart.series.i[].silent.tooltip.textStyle.textShadowColor .....	4386
6.19.1321 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetX.....	4387
6.19.1322 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetY .....	4387
6.19.1323 chart.series.i[].silent.tooltip.textStyle.width .....	4387
6.19.1324 chart.series.i[].singleAxisIndex.....	4388
6.19.1325 chart.series.i[].smooth.....	4388
6.19.1326 chart.series.i[].smoothMonotone .....	4389
Erlaubte Werte .....	4389
6.19.1327 chart.series.i[].sort.....	4389
6.19.1328 chart.series.i[].splitLine.length .....	4390
Erlaubte Werte .....	4390
6.19.1329 chart.series.i[].splitLine.lineStyle.color .....	4390
6.19.1330 chart.series.i[].splitLine.lineStyle.object.....	4392



6.19.1331	chart.series.i[].splitLine.lineStyle.opacity .....	4392
6.19.1332	chart.series.i[].splitLine.lineStyle.shadowBlur .....	4392
6.19.1333	chart.series.i[].splitLine.lineStyle.shadowColor .....	4393
6.19.1334	chart.series.i[].splitLine.lineStyle.shadowOffsetX .....	4393
6.19.1335	chart.series.i[].splitLine.lineStyle.shadowOffsetY .....	4393
6.19.1336	chart.series.i[].splitLine.lineStyle.type .....	4394
	Erlaubte Werte .....	4394
6.19.1337	chart.series.i[].splitLine.lineStyle.width .....	4394
6.19.1338	chart.series.i[].splitLine.object.....	4395
6.19.1339	chart.series.i[].splitLine.show .....	4395
6.19.1340	chart.series.i[].splitNumber.....	4395
6.19.1341	chart.series.i[].squareRatio .....	4396
6.19.1342	chart.series.i[].stack.....	4396
6.19.1343	chart.series.i[].startAngle .....	4396
6.19.1344	chart.series.i[].step .....	4397
	Erlaubte Werte .....	4397
6.19.1345	chart.series.i[].stillShowZeroSum.....	4397
6.19.1346	chart.series.i[].symbol .....	4398
	Erlaubte Werte .....	4398
6.19.1347	chart.series.i[].symbolBoundingData .....	4398
6.19.1348	chart.series.i[].symbolClip.....	4400
6.19.1349	chart.series.i[].symbolKeepAspect .....	4400
6.19.1350	chart.series.i[].symbolMargin.....	4401
6.19.1351	chart.series.i[].symbolOffset .....	4402
	Erlaubte Werte .....	4402
6.19.1352	chart.series.i[].symbolPatternSize.....	4402
6.19.1353	chart.series.i[].symbolRepeat .....	4403
	Erlaubte Werte .....	4404
6.19.1354	chart.series.i[].symbolRepeatDirection .....	4405
	Erlaubte Werte .....	4405
6.19.1355	chart.series.i[].symbolRotate .....	4406
6.19.1356	chart.series.i[].symbolSize .....	4406
6.19.1357	chart.series.i[].title.backgroundColor.....	4407
6.19.1358	chart.series.i[].title.borderColor .....	4408
6.19.1359	chart.series.i[].title.borderRadius.....	4408
6.19.1360	chart.series.i[].title.borderWidth.....	4409

6.19.1361	chart.series.i[].title.color .....	4409
6.19.1362	chart.series.i[].title.fontFamily .....	4409
6.19.1363	chart.series.i[].title.fontSize .....	4410
6.19.1364	chart.series.i[].title.fontStyle .....	4410
	Erlaubte Werte .....	4411
6.19.1365	chart.series.i[].title.fontWeight .....	4411
	Erlaubte Werte .....	4411
6.19.1366	chart.series.i[].title.height .....	4411
6.19.1367	chart.series.i[].title.lineHeight .....	4412
6.19.1368	chart.series.i[].title.object .....	4412
6.19.1369	chart.series.i[].title.offsetCenter .....	4413
	Erlaubte Werte .....	4413
6.19.1370	chart.series.i[].title.padding .....	4413
6.19.1371	chart.series.i[].title.rich.object .....	4414
6.19.1372	chart.series.i[].title.shadowBlur .....	4415
6.19.1373	chart.series.i[].title.shadowColor .....	4415
6.19.1374	chart.series.i[].title.shadowOffsetX .....	4415
6.19.1375	chart.series.i[].title.shadowOffsetY .....	4416
6.19.1376	chart.series.i[].title.show .....	4416
6.19.1377	chart.series.i[].title.textBorderColor .....	4416
6.19.1378	chart.series.i[].title.textBorderWidth .....	4417
6.19.1379	chart.series.i[].title.textShadowBlur .....	4417
6.19.1380	chart.series.i[].title.textShadowColor .....	4417
6.19.1381	chart.series.i[].title.textShadowOffsetX .....	4418
6.19.1382	chart.series.i[].title.textShadowOffsetY .....	4418
6.19.1383	chart.series.i[].title.width .....	4419
6.19.1384	chart.series.i[].tooltip.backgroundColor .....	4419
6.19.1385	chart.series.i[].tooltip.borderColor .....	4419
6.19.1386	chart.series.i[].tooltip.borderWidth .....	4420
6.19.1387	chart.series.i[].tooltip.extraCssText .....	4420
6.19.1388	chart.series.i[].tooltip.formatter .....	4421
6.19.1389	chart.series.i[].tooltip.object .....	4423
6.19.1390	chart.series.i[].tooltip.padding .....	4423
6.19.1391	chart.series.i[].tooltip.position .....	4424
6.19.1392	chart.series.i[].tooltip.textStyle.color .....	4425
6.19.1393	chart.series.i[].tooltip.textStyle.fontFamily .....	4426

6.19.1394	chart.series.i[].tooltip.textStyle.fontSize.....	4426
6.19.1395	chart.series.i[].tooltip.textStyle.fontStyle .....	4427
	Erlaubte Werte .....	4427
6.19.1396	chart.series.i[].tooltip.textStyle.fontWeight.....	4427
	Erlaubte Werte .....	4427
6.19.1397	chart.series.i[].tooltip.textStyle.height.....	4428
6.19.1398	chart.series.i[].tooltip.textStyle.lineHeight .....	4428
6.19.1399	chart.series.i[].tooltip.textStyle.object.....	4429
6.19.1400	chart.series.i[].tooltip.textStyle.textBorderColor .....	4429
6.19.1401	chart.series.i[].tooltip.textStyle.textBorderWidth .....	4429
6.19.1402	chart.series.i[].tooltip.textStyle.textShadowBlur .....	4430
6.19.1403	chart.series.i[].tooltip.textStyle.textShadowColor .....	4430
6.19.1404	chart.series.i[].tooltip.textStyle.textShadowOffsetX .....	4431
6.19.1405	chart.series.i[].tooltip.textStyle.textShadowOffsetY .....	4431
6.19.1406	chart.series.i[].tooltip.textStyle.width.....	4431
6.19.1407	chart.series.i[].top.....	4432
	Erlaubte Werte .....	4432
6.19.1408	chart.series.i[].upperLabel.align .....	4432
	Erlaubte Werte .....	4433
6.19.1409	chart.series.i[].upperLabel.backgroundColor .....	4433
6.19.1410	chart.series.i[].upperLabel.borderColor .....	4434
6.19.1411	chart.series.i[].upperLabel.borderRadius.....	4434
6.19.1412	chart.series.i[].upperLabel.borderWidth .....	4435
6.19.1413	chart.series.i[].upperLabel.color.....	4435
6.19.1414	chart.series.i[].upperLabel.distance .....	4435
6.19.1415	chart.series.i[].upperLabel.ellipsis.....	4436
6.19.1416	chart.series.i[].upperLabel.fontFamily .....	4436
6.19.1417	chart.series.i[].upperLabel.fontSize.....	4436
6.19.1418	chart.series.i[].upperLabel.fontStyle .....	4437
	Erlaubte Werte .....	4437
6.19.1419	chart.series.i[].upperLabel.fontWeight.....	4437
	Erlaubte Werte .....	4438
6.19.1420	chart.series.i[].upperLabel.formatter .....	4438
6.19.1421	chart.series.i[].upperLabel.height.....	4439
6.19.1422	chart.series.i[].upperLabel.lineHeight.....	4439
6.19.1423	chart.series.i[].upperLabel.object.....	4440

6.19.1424	chart.series.i[].upperLabel.offset.....	4441
6.19.1425	chart.series.i[].upperLabel.padding .....	4441
6.19.1426	chart.series.i[].upperLabel.position.....	4441
	Erlaubte Werte .....	4442
6.19.1427	chart.series.i[].upperLabel.rich.object .....	4442
6.19.1428	chart.series.i[].upperLabel.rotate .....	4443
6.19.1429	chart.series.i[].upperLabel.shadowBlur .....	4444
6.19.1430	chart.series.i[].upperLabel.shadowColor .....	4444
6.19.1431	chart.series.i[].upperLabel.shadowOffsetX .....	4444
6.19.1432	chart.series.i[].upperLabel.shadowOffsetY .....	4445
6.19.1433	chart.series.i[].upperLabel.show .....	4445
6.19.1434	chart.series.i[].upperLabel.textBorderColor .....	4445
6.19.1435	chart.series.i[].upperLabel.textBorderWidth .....	4446
6.19.1436	chart.series.i[].upperLabel.textShadowBlur .....	4446
6.19.1437	chart.series.i[].upperLabel.textShadowColor .....	4446
6.19.1438	chart.series.i[].upperLabel.textShadowOffsetX .....	4447
6.19.1439	chart.series.i[].upperLabel.textShadowOffsetY .....	4447
6.19.1440	chart.series.i[].upperLabel.verticalAlign .....	4448
	Erlaubte Werte .....	4448
6.19.1441	chart.series.i[].upperLabel.width.....	4448
6.19.1442	chart.series.i[].visibleMin.....	4449
6.19.1443	chart.series.i[].visualDimension.....	4449
6.19.1444	chart.series.i[].visualMax .....	4450
6.19.1445	chart.series.i[].visualMin.....	4450
6.19.1446	chart.series.i[].width .....	4451
6.19.1447	chart.series.i[].xAxisIndex.....	4451
6.19.1448	chart.series.i[].yAxisIndex.....	4451
6.19.1449	chart.series.i[].z .....	4452
6.19.1450	chart.series.i[].zlevel .....	4452
6.19.1451	chart.series.i[].zoom .....	4453
6.19.1452	chart.series.i[].zoomToNodeRatio .....	4453
6.19.1453	chart.series.i[].silent.children .....	4453
6.19.1454	chart.series.fixed.name .....	4454
6.19.1455	chart.series.i[].name .....	4454
6.19.1456	chart.series.i[].label.alignTo.....	4455
	Erlaubte Werte .....	4455

6.19.1457	chart.series.i[].label.edgeDistance .....	4455
6.19.1458	chart.series.i[].label.bleedMargin .....	4456
6.19.1459	chart.series.i[].label.distanceToLabelLine .....	4456
6.19.1460	chart.series.i[].minShowLabelAngle .....	4456
6.20	<b>ECharts (5) Single Axis.....</b>	<b>4457</b>
6.20.1	chart.singleAxis.object.....	4457
6.20.2	chart.singleAxis.id.....	4457
6.20.3	chart.singleAxis.zlevel.....	4457
6.20.4	chart.singleAxis.z .....	4458
6.20.5	chart.singleAxis.left.....	4458
	Erlaubte Werte .....	4459
6.20.6	chart.singleAxis.top .....	4459
	Erlaubte Werte .....	4459
6.20.7	chart.singleAxis.right .....	4460
	Erlaubte Werte .....	4460
6.20.8	chart.singleAxis.bottom.....	4460
	Erlaubte Werte .....	4460
6.20.9	chart.singleAxis.width.....	4461
6.20.10	chart.singleAxis.height .....	4461
6.20.11	chart.singleAxis.orient .....	4461
	Erlaubte Werte .....	4462
6.20.12	chart.singleAxis.type.....	4462
	Erlaubte Werte .....	4462
6.20.13	chart.singleAxis.name.....	4462
6.20.14	chart.singleAxis.nameLocation .....	4463
	Erlaubte Werte .....	4463
6.20.15	chart.singleAxis.nameTextStyle.object .....	4463
6.20.16	chart.singleAxis.nameTextStyle.color .....	4464
6.20.17	chart.singleAxis.nameTextStyle.fontStyle.....	4464
	Erlaubte Werte .....	4464
6.20.18	chart.singleAxis.nameTextStyle.fontWeight .....	4464
	Erlaubte Werte .....	4465
6.20.19	chart.singleAxis.nameTextStyle.fontFamily.....	4465
6.20.20	chart.singleAxis.nameTextStyle.fontSize .....	4465
6.20.21	chart.singleAxis.nameTextStyle.align.....	4466
	Erlaubte Werte .....	4466

6.20.22	chart.singleAxis.nameTextStyle.verticalAlign .....	4466
	Erlaubte Werte .....	4467
6.20.23	chart.singleAxis.nameTextStyle.lineHeight.....	4467
6.20.24	chart.singleAxis.nameTextStyle.backgroundColor.....	4468
6.20.25	chart.singleAxis.nameTextStyle.borderColor .....	4468
6.20.26	chart.singleAxis.nameTextStyle.borderWidth.....	4469
6.20.27	chart.singleAxis.nameTextStyle.borderRadius .....	4469
6.20.28	chart.singleAxis.nameTextStyle.padding .....	4469
6.20.29	chart.singleAxis.nameTextStyle.shadowColor.....	4470
6.20.30	chart.singleAxis.nameTextStyle.shadowBlur.....	4470
6.20.31	chart.singleAxis.nameTextStyle.shadowOffsetX.....	4471
6.20.32	chart.singleAxis.nameTextStyle.shadowOffsetY.....	4471
6.20.33	chart.singleAxis.nameTextStyle.width .....	4471
6.20.34	chart.singleAxis.nameTextStyle.height .....	4472
6.20.35	chart.singleAxis.nameTextStyle.textBorderColor.....	4472
6.20.36	chart.singleAxis.nameTextStyle.textBorderWidth .....	4473
6.20.37	chart.singleAxis.nameTextStyle.textShadowColor.....	4473
6.20.38	chart.singleAxis.nameTextStyle.textShadowBlur.....	4473
6.20.39	chart.singleAxis.nameTextStyle.textShadowOffsetX.....	4474
6.20.40	chart.singleAxis.nameTextStyle.textShadowOffsetY.....	4474
6.20.41	chart.singleAxis.nameTextStyle.rich.object.....	4474
6.20.42	chart.singleAxis.nameGap.....	4475
6.20.43	chart.singleAxis.nameRotate .....	4476
6.20.44	chart.singleAxis.inverse .....	4476
6.20.45	chart.singleAxis.boundaryGap .....	4476
6.20.46	chart.singleAxis.min.....	4477
6.20.47	chart.singleAxis.max .....	4478
6.20.48	chart.singleAxis.scale.....	4478
6.20.49	chart.singleAxis.splitNumber .....	4478
6.20.50	chart.singleAxis.minInterval.....	4479
6.20.51	chart.singleAxis.interval .....	4479
6.20.52	chart.singleAxis.logBase .....	4480
6.20.53	chart.singleAxis.silent.....	4480
6.20.54	chart.singleAxis.triggerEvent .....	4481
6.20.55	chart.singleAxis.axisLine.object.....	4481
6.20.56	chart.singleAxis.axisLine.show.....	4482

6.20.57	chart.singleAxis.axisLine.symbol .....	4482
	Erlaubte Werte .....	4482
6.20.58	chart.singleAxis.axisLine.symbolSize .....	4482
6.20.59	chart.singleAxis.axisLine.symbolOffset .....	4483
6.20.60	chart.singleAxis.axisLine.lineStyle.object .....	4483
6.20.61	chart.singleAxis.axisLine.lineStyle.color .....	4484
6.20.62	chart.singleAxis.axisLine.lineStyle.width .....	4485
6.20.63	chart.singleAxis.axisLine.lineStyle.type .....	4486
	Erlaubte Werte .....	4486
6.20.64	chart.singleAxis.axisLine.lineStyle.shadowBlur .....	4486
6.20.65	chart.singleAxis.axisLine.lineStyle.shadowColor .....	4487
6.20.66	chart.singleAxis.axisLine.lineStyle.shadowOffsetX .....	4487
6.20.67	chart.singleAxis.axisLine.lineStyle.shadowOffsetY .....	4487
6.20.68	chart.singleAxis.axisLine.lineStyle.opacity .....	4488
6.20.69	chart.singleAxis.axisTick.object .....	4488
6.20.70	chart.singleAxis.axisTick.show .....	4488
6.20.71	chart.singleAxis.axisTick.alignWithLabel .....	4489
6.20.72	chart.singleAxis.axisTick.interval .....	4489
6.20.73	chart.singleAxis.axisTick.inside .....	4490
6.20.74	chart.singleAxis.axisTick.length .....	4490
6.20.75	chart.singleAxis.axisTick.lineStyle.object .....	4490
6.20.76	chart.singleAxis.axisTick.lineStyle.color .....	4491
6.20.77	chart.singleAxis.axisTick.lineStyle.width .....	4491
6.20.78	chart.singleAxis.axisTick.lineStyle.type .....	4491
	Erlaubte Werte .....	4492
6.20.79	chart.singleAxis.axisTick.lineStyle.shadowBlur .....	4492
6.20.80	chart.singleAxis.axisTick.lineStyle.shadowColor .....	4492
6.20.81	chart.singleAxis.axisTick.lineStyle.shadowOffsetX .....	4493
6.20.82	chart.singleAxis.axisTick.lineStyle.shadowOffsetY .....	4493
6.20.83	chart.singleAxis.axisTick.lineStyle.opacity .....	4493
6.20.84	chart.singleAxis.axisLabel.object .....	4494
6.20.85	chart.singleAxis.axisLabel.show .....	4494
6.20.86	chart.singleAxis.axisLabel.interval .....	4494
6.20.87	chart.singleAxis.axisLabel.inside .....	4495
6.20.88	chart.singleAxis.axisLabel.rotate .....	4495
6.20.89	chart.singleAxis.axisLabel.margin .....	4495

6.20.90	chart.singleAxis.axisLabel.formatter .....	4496
6.20.91	chart.singleAxis.axisLabel.showMinLabel .....	4496
6.20.92	chart.singleAxis.axisLabel.showMaxLabel.....	4497
6.20.93	chart.singleAxis.axisLabel.color .....	4497
6.20.94	chart.singleAxis.axisLabel.fontStyle .....	4498
	Erlaubte Werte .....	4498
6.20.95	chart.singleAxis.axisLabel.fontWeight.....	4498
	Erlaubte Werte .....	4499
6.20.96	chart.singleAxis.axisLabel.fontFamily .....	4499
6.20.97	chart.singleAxis.axisLabel.fontSize.....	4499
6.20.98	chart.singleAxis.axisLabel.align .....	4500
	Erlaubte Werte .....	4500
6.20.99	chart.singleAxis.axisLabel.verticalAlign.....	4500
	Erlaubte Werte .....	4501
6.20.100	chart.singleAxis.axisLabel.lineHeight .....	4501
6.20.101	chart.singleAxis.axisLabel.backgroundColor .....	4502
6.20.102	chart.singleAxis.axisLabel.borderColor .....	4502
6.20.103	chart.singleAxis.axisLabel.borderWidth .....	4503
6.20.104	chart.singleAxis.axisLabel.borderRadius.....	4503
6.20.105	chart.singleAxis.axisLabel.padding.....	4503
6.20.106	chart.singleAxis.axisLabel.shadowColor .....	4504
6.20.107	chart.singleAxis.axisLabel.shadowBlur .....	4504
6.20.108	chart.singleAxis.axisLabel.shadowOffsetX .....	4505
6.20.109	chart.singleAxis.axisLabel.shadowOffsetY .....	4505
6.20.110	chart.singleAxis.axisLabel.width.....	4505
6.20.111	chart.singleAxis.axisLabel.height.....	4506
6.20.112	chart.singleAxis.axisLabel.textBorderColor .....	4506
6.20.113	chart.singleAxis.axisLabel.textBorderWidth .....	4507
6.20.114	chart.singleAxis.axisLabel.textShadowColor .....	4507
6.20.115	chart.singleAxis.axisLabel.textShadowBlur .....	4507
6.20.116	chart.singleAxis.axisLabel.textShadowOffsetX .....	4508
6.20.117	chart.singleAxis.axisLabel.textShadowOffsetY .....	4508
6.20.118	chart.singleAxis.axisLabel.rich.object .....	4508
6.20.119	chart.singleAxis.splitLine.object .....	4509
6.20.120	chart.singleAxis.splitLine.show .....	4510
6.20.121	chart.singleAxis.splitLine.interval.....	4510



6.20.122	chart.singleAxis.splitLine.lineStyle.object.....	4510
6.20.123	chart.singleAxis.splitLine.lineStyle.color.....	4511
6.20.124	chart.singleAxis.splitLine.lineStyle.width.....	4511
6.20.125	chart.singleAxis.splitLine.lineStyle.type.....	4512
	Erlaubte Werte .....	4512
6.20.126	chart.singleAxis.splitLine.lineStyle.shadowBlur .....	4512
6.20.127	chart.singleAxis.splitLine.lineStyle.shadowColor .....	4513
6.20.128	chart.singleAxis.splitLine.lineStyle.shadowOffsetX.....	4513
6.20.129	chart.singleAxis.splitLine.lineStyle.shadowOffsetY .....	4513
6.20.130	chart.singleAxis.splitLine.lineStyle.opacity.....	4514
6.20.131	chart.singleAxis.splitArea.object.....	4514
6.20.132	chart.singleAxis.splitArea.interval .....	4514
6.20.133	chart.singleAxis.splitArea.show .....	4515
6.20.134	chart.singleAxis.splitArea.areaStyle.object.....	4515
6.20.135	chart.singleAxis.splitArea.areaStyle.color.....	4515
6.20.136	chart.singleAxis.splitArea.areaStyle.shadowBlur .....	4516
6.20.137	chart.singleAxis.splitArea.areaStyle.shadowColor .....	4516
6.20.138	chart.singleAxis.splitArea.areaStyle.shadowOffsetX .....	4517
6.20.139	chart.singleAxis.splitArea.areaStyle.shadowOffsetY .....	4517
6.20.140	chart.singleAxis.splitArea.areaStyle.opacity.....	4517
6.20.141	chart.singleAxis.axisPointer.object.....	4518
6.20.142	chart.singleAxis.axisPointer.show .....	4518
6.20.143	chart.singleAxis.axisPointer.type.....	4519
	Erlaubte Werte .....	4519
6.20.144	chart.singleAxis.axisPointer.snap .....	4519
6.20.145	chart.singleAxis.axisPointer.z.....	4519
6.20.146	chart.singleAxis.axisPointer.label.object .....	4520
6.20.147	chart.singleAxis.axisPointer.label.show .....	4520
6.20.148	chart.singleAxis.axisPointer.label.precision.....	4521
6.20.149	chart.singleAxis.axisPointer.label.formatter.....	4521
6.20.150	chart.singleAxis.axisPointer.label.margin .....	4522
6.20.151	chart.singleAxis.axisPointer.label.color .....	4523
6.20.152	chart.singleAxis.axisPointer.label.fontStyle.....	4523
	Erlaubte Werte .....	4523
6.20.153	chart.singleAxis.axisPointer.label.fontWeight .....	4524
	Erlaubte Werte .....	4524

6.20.154	chart.singleAxis.axisPointer.label.fontFamily .....	4524
6.20.155	chart.singleAxis.axisPointer.label.fontSize .....	4525
6.20.156	chart.singleAxis.axisPointer.label.lineHeight.....	4525
6.20.157	chart.singleAxis.axisPointer.label.width .....	4526
6.20.158	chart.singleAxis.axisPointer.label.height .....	4526
6.20.159	chart.singleAxis.axisPointer.label.textBorderColor .....	4526
6.20.160	chart.singleAxis.axisPointer.label.textBorderWidth .....	4527
6.20.161	chart.singleAxis.axisPointer.label.textShadowColor .....	4527
6.20.162	chart.singleAxis.axisPointer.label.textShadowBlur .....	4528
6.20.163	chart.singleAxis.axisPointer.label.textShadowOffsetX.....	4528
6.20.164	chart.singleAxis.axisPointer.label.textShadowOffsetY .....	4528
6.20.165	chart.singleAxis.axisPointer.label.padding .....	4529
6.20.166	chart.singleAxis.axisPointer.label.backgroundColor.....	4529
6.20.167	chart.singleAxis.axisPointer.label.borderColor.....	4530
6.20.168	chart.singleAxis.axisPointer.label.borderWidth.....	4530
6.20.169	chart.singleAxis.axisPointer.label.shadowBlur .....	4530
6.20.170	chart.singleAxis.axisPointer.label.shadowColor .....	4531
6.20.171	chart.singleAxis.axisPointer.label.shadowOffsetX.....	4531
6.20.172	chart.singleAxis.axisPointer.label.shadowOffsetY .....	4532
6.20.173	chart.singleAxis.axisPointer.lineStyle.object .....	4532
6.20.174	chart.singleAxis.axisPointer.lineStyle.color .....	4532
6.20.175	chart.singleAxis.axisPointer.lineStyle.width .....	4534
6.20.176	chart.singleAxis.axisPointer.lineStyle.type .....	4534
	Erlaubte Werte .....	4534
6.20.177	chart.singleAxis.axisPointer.lineStyle.shadowBlur.....	4534
6.20.178	chart.singleAxis.axisPointer.lineStyle.shadowColor .....	4535
6.20.179	chart.singleAxis.axisPointer.lineStyle.shadowOffsetX .....	4535
6.20.180	chart.singleAxis.axisPointer.lineStyle.shadowOffsetY.....	4536
6.20.181	chart.singleAxis.axisPointer.lineStyle.opacity .....	4536
6.20.182	chart.singleAxis.axisPointer.shadowStyle.object .....	4536
6.20.183	chart.singleAxis.axisPointer.shadowStyle.color .....	4537
6.20.184	chart.singleAxis.axisPointer.shadowStyle.shadowBlur.....	4538
6.20.185	chart.singleAxis.axisPointer.shadowStyle.shadowColor .....	4539
6.20.186	chart.singleAxis.axisPointer.shadowStyle.shadowOffsetX .....	4539
6.20.187	chart.singleAxis.axisPointer.shadowStyle.shadowOffsetY.....	4540
6.20.188	chart.singleAxis.axisPointer.shadowStyle.opacity .....	4540

6.20.189	chart.singleAxis.axisPointer.triggerTooltip .....	4540
6.20.190	chart.singleAxis.axisPointer.value .....	4541
6.20.191	chart.singleAxis.axisPointer.status .....	4541
	Erlaubte Werte .....	4541
6.20.192	chart.singleAxis.axisPointer.handle.object .....	4541
6.20.193	chart.singleAxis.axisPointer.handle.show .....	4542
6.20.194	chart.singleAxis.axisPointer.handle.icon .....	4542
6.20.195	chart.singleAxis.axisPointer.handle.size .....	4543
6.20.196	chart.singleAxis.axisPointer.handle.margin .....	4543
6.20.197	chart.singleAxis.axisPointer.handle.color .....	4543
6.20.198	chart.singleAxis.axisPointer.handle.throttle .....	4544
6.20.199	chart.singleAxis.axisPointer.handle.shadowBlur .....	4544
6.20.200	chart.singleAxis.axisPointer.handle.shadowColor .....	4545
6.20.201	chart.singleAxis.axisPointer.handle.shadowOffsetX .....	4545
6.20.202	chart.singleAxis.axisPointer.handle.shadowOffsetY .....	4545
6.20.203	chart.singleAxis.tooltip.object .....	4546
6.20.204	chart.singleAxis.tooltip.show .....	4546
6.20.205	chart.singleAxis.tooltip.trigger .....	4547
	Erlaubte Werte .....	4547
6.20.206	chart.singleAxis.tooltip.axisPointer.object .....	4547
6.20.207	chart.singleAxis.tooltip.axisPointer.type .....	4548
	Erlaubte Werte .....	4548
6.20.208	chart.singleAxis.tooltip.axisPointer.axis .....	4548
	Erlaubte Werte .....	4549
6.20.209	chart.singleAxis.tooltip.axisPointer.snap .....	4549
6.20.210	chart.singleAxis.tooltip.axisPointer.z .....	4549
6.20.211	chart.singleAxis.tooltip.axisPointer.label.object .....	4550
6.20.212	chart.singleAxis.tooltip.axisPointer.label.show .....	4550
6.20.213	chart.singleAxis.tooltip.axisPointer.label.precision .....	4550
6.20.214	chart.singleAxis.tooltip.axisPointer.label.formatter .....	4551
6.20.215	chart.singleAxis.tooltip.axisPointer.label.margin .....	4552
6.20.216	chart.singleAxis.tooltip.axisPointer.label.color .....	4553
6.20.217	chart.singleAxis.tooltip.axisPointer.label.fontStyle .....	4553
	Erlaubte Werte .....	4553
6.20.218	chart.singleAxis.tooltip.axisPointer.label.fontWeight .....	4554
	Erlaubte Werte .....	4554

6.20.219	chart.singleAxis.tooltip.axisPointer.label.fontFamily.....	4554
6.20.220	chart.singleAxis.tooltip.axisPointer.label.fontSize .....	4555
6.20.221	chart.singleAxis.tooltip.axisPointer.label.lineHeight .....	4555
6.20.222	chart.singleAxis.tooltip.axisPointer.label.width .....	4556
6.20.223	chart.singleAxis.tooltip.axisPointer.label.height .....	4556
6.20.224	chart.singleAxis.tooltip.axisPointer.label.textBorderColor.....	4556
6.20.225	chart.singleAxis.tooltip.axisPointer.label.textBorderWidth.....	4557
6.20.226	chart.singleAxis.tooltip.axisPointer.label.textShadowColor .....	4557
6.20.227	chart.singleAxis.tooltip.axisPointer.label.textShadowBlur.....	4558
6.20.228	chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetX .....	4558
6.20.229	chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetY.....	4558
6.20.230	chart.singleAxis.tooltip.axisPointer.label.padding.....	4559
6.20.231	chart.singleAxis.tooltip.axisPointer.label.backgroundColor .....	4559
6.20.232	chart.singleAxis.tooltip.axisPointer.label.borderColor .....	4560
6.20.233	chart.singleAxis.tooltip.axisPointer.label.borderWidth .....	4560
6.20.234	chart.singleAxis.tooltip.axisPointer.label.shadowBlur.....	4560
6.20.235	chart.singleAxis.tooltip.axisPointer.label.shadowColor .....	4561
6.20.236	chart.singleAxis.tooltip.axisPointer.label.shadowOffsetX .....	4561
6.20.237	chart.singleAxis.tooltip.axisPointer.label.shadowOffsetY.....	4562
6.20.238	chart.singleAxis.tooltip.axisPointer.lineStyle.object.....	4562
6.20.239	chart.singleAxis.tooltip.axisPointer.lineStyle.color.....	4562
6.20.240	chart.singleAxis.tooltip.axisPointer.lineStyle.width.....	4564
6.20.241	chart.singleAxis.tooltip.axisPointer.lineStyle.type .....	4564
	Erlaubte Werte .....	4564
6.20.242	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowBlur .....	4564
6.20.243	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowColor .....	4565
6.20.244	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetX .....	4565
6.20.245	chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetY .....	4566
6.20.246	chart.singleAxis.tooltip.axisPointer.lineStyle.opacity .....	4566
6.20.247	chart.singleAxis.tooltip.axisPointer.shadowStyle.object.....	4566
6.20.248	chart.singleAxis.tooltip.axisPointer.shadowStyle.color.....	4567
6.20.249	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowBlur .....	4568
6.20.250	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowColor .....	4569
6.20.251	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetX .....	4569
6.20.252	chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetY .....	4570
6.20.253	chart.singleAxis.tooltip.axisPointer.shadowStyle.opacity .....	4570

6.20.254	chart.singleAxis.tooltip.axisPointer.crossStyle.object .....	4570
6.20.255	chart.singleAxis.tooltip.axisPointer.crossStyle.color .....	4571
6.20.256	chart.singleAxis.tooltip.axisPointer.crossStyle.width .....	4572
6.20.257	chart.singleAxis.tooltip.axisPointer.crossStyle.type .....	4573
	Erlaubte Werte .....	4573
6.20.258	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowBlur.....	4573
6.20.259	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowColor.....	4574
6.20.260	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetX.....	4574
6.20.261	chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetY.....	4574
6.20.262	chart.singleAxis.tooltip.axisPointer.crossStyle.opacity .....	4575
6.20.263	chart.singleAxis.tooltip.position .....	4575
6.20.264	chart.singleAxis.tooltip.formatter.....	4577
6.20.265	chart.singleAxis.tooltip.backgroundColor.....	4579
6.20.266	chart.singleAxis.tooltip.borderColor .....	4579
6.20.267	chart.singleAxis.tooltip.borderWidth.....	4580
6.20.268	chart.singleAxis.tooltip.padding .....	4580
6.20.269	chart.singleAxis.tooltip.textStyle.object .....	4581
6.20.270	chart.singleAxis.tooltip.textStyle.color .....	4581
6.20.271	chart.singleAxis.tooltip.textStyle.fontStyle.....	4581
	Erlaubte Werte .....	4582
6.20.272	chart.singleAxis.tooltip.textStyle.fontWeight .....	4582
	Erlaubte Werte .....	4582
6.20.273	chart.singleAxis.tooltip.textStyle.fontFamily.....	4582
6.20.274	chart.singleAxis.tooltip.textStyle.fontSize .....	4583
6.20.275	chart.singleAxis.tooltip.textStyle.lineHeight.....	4583
6.20.276	chart.singleAxis.tooltip.textStyle.width .....	4584
6.20.277	chart.singleAxis.tooltip.textStyle.height .....	4584
6.20.278	chart.singleAxis.tooltip.textStyle.textBorderColor .....	4585
6.20.279	chart.singleAxis.tooltip.textStyle.textBorderWidth .....	4585
6.20.280	chart.singleAxis.tooltip.textStyle.textShadowColor.....	4585
6.20.281	chart.singleAxis.tooltip.textStyle.textShadowBlur .....	4586
6.20.282	chart.singleAxis.tooltip.textStyle.textShadowOffsetX.....	4586
6.20.283	chart.singleAxis.tooltip.textStyle.textShadowOffsetY .....	4586
6.20.284	chart.singleAxis.tooltip.extraCssText .....	4587
6.21	ECharts (5) Text Style.....	4587
6.21.1	chart.textStyle.object .....	4587

6.21.2	chart.textStyle.color .....	4588
6.21.3	chart.textStyle.fontStyle.....	4588
	Erlaubte Werte .....	4588
6.21.4	chart.textStyle.fontWeight .....	4588
	Erlaubte Werte .....	4589
6.21.5	chart.textStyle.fontFamily.....	4589
6.21.6	chart.textStyle.fontSize .....	4589
6.21.7	chart.textStyle.lineHeight .....	4590
6.21.8	chart.textStyle.width .....	4590
6.21.9	chart.textStyle.height .....	4591
6.21.10	chart.textStyle.textBorderColor.....	4591
6.21.11	chart.textStyle.textBorderWidth.....	4592
6.21.12	chart.textStyle.textShadowColor.....	4592
6.21.13	chart.textStyle.textShadowBlur.....	4592
6.21.14	chart.textStyle.textShadowOffsetX.....	4593
6.21.15	chart.textStyle.textShadowOffsetY.....	4593
6.21.16	chart.textStyle.overflow .....	4593
	Erlaubte Werte .....	4594
6.21.17	chart.textStyle.ellipsis .....	4594
6.22	<b>ECharts (5) Timeline.....</b>	<b>4594</b>
6.22.1	chart.timeline.object .....	4594
6.22.2	chart.timeline.show .....	4597
6.22.3	chart.timeline.type .....	4598
6.22.4	chart.timeline.axisType .....	4598
	Erlaubte Werte .....	4598
6.22.5	chart.timeline.currentIndex .....	4599
6.22.6	chart.timeline.autoPlay .....	4599
6.22.7	chart.timeline.rewind .....	4599
6.22.8	chart.timeline.loop .....	4600
6.22.9	chart.timeline.playInterval.....	4600
6.22.10	chart.timeline.realtime.....	4600
6.22.11	chart.timeline.controlPosition.....	4601
	Erlaubte Werte .....	4601
6.22.12	chart.timeline.zlevel .....	4601
6.22.13	chart.timeline.z .....	4602
6.22.14	chart.timeline.left .....	4602

	Erlaubte Werte .....	4603
6.22.15	chart.timeline.top .....	4603
	Erlaubte Werte .....	4603
6.22.16	chart.timeline.right .....	4603
	Erlaubte Werte .....	4604
6.22.17	chart.timeline.bottom .....	4604
	Erlaubte Werte .....	4604
6.22.18	chart.timeline.padding .....	4604
6.22.19	chart.timeline.orient .....	4605
	Erlaubte Werte .....	4605
6.22.20	chart.timeline.inverse .....	4606
6.22.21	chart.timeline.symbol .....	4606
	Erlaubte Werte .....	4606
6.22.22	chart.timeline.symbolSize .....	4607
6.22.23	chart.timeline.symbolRotate .....	4607
6.22.24	chart.timeline.symbolKeepAspect .....	4607
6.22.25	chart.timeline.symbolOffset .....	4608
6.22.26	chart.timeline.lineStyle.object .....	4608
6.22.27	chart.timeline.lineStyle.show .....	4609
6.22.28	chart.timeline.lineStyle.color .....	4609
6.22.29	chart.timeline.lineStyle.width .....	4610
6.22.30	chart.timeline.lineStyle.type .....	4611
	Erlaubte Werte .....	4611
6.22.31	chart.timeline.lineStyle.shadowBlur .....	4611
6.22.32	chart.timeline.lineStyle.shadowColor .....	4612
6.22.33	chart.timeline.lineStyle.shadowOffsetX .....	4612
6.22.34	chart.timeline.lineStyle.shadowOffsetY .....	4612
6.22.35	chart.timeline.lineStyle.opacity .....	4613
6.22.36	chart.timeline.label.object .....	4613
6.22.37	chart.timeline.label.position .....	4613
	Erlaubte Werte .....	4614
6.22.38	chart.timeline.label.show .....	4614
6.22.39	chart.timeline.label.interval .....	4615
6.22.40	chart.timeline.label.rotate .....	4615
6.22.41	chart.timeline.label.formatter .....	4615
6.22.42	chart.timeline.label.color .....	4616

6.22.43	chart.timeline.label.fontStyle .....	4616
	Erlaubte Werte .....	4617
6.22.44	chart.timeline.label.fontWeight .....	4617
	Erlaubte Werte .....	4617
6.22.45	chart.timeline.label.fontFamily .....	4617
6.22.46	chart.timeline.label.fontSize .....	4618
6.22.47	chart.timeline.label.align .....	4618
	Erlaubte Werte .....	4619
6.22.48	chart.timeline.label.verticalAlign .....	4619
	Erlaubte Werte .....	4619
6.22.49	chart.timeline.label.lineHeight .....	4620
6.22.50	chart.timeline.label.backgroundColor .....	4620
6.22.51	chart.timeline.label.borderColor .....	4621
6.22.52	chart.timeline.label.borderWidth .....	4621
6.22.53	chart.timeline.label.borderRadius .....	4622
6.22.54	chart.timeline.label.padding .....	4622
6.22.55	chart.timeline.label.shadowColor .....	4622
6.22.56	chart.timeline.label.shadowBlur .....	4623
6.22.57	chart.timeline.label.shadowOffsetX .....	4623
6.22.58	chart.timeline.label.shadowOffsetY .....	4623
6.22.59	chart.timeline.label.width .....	4624
6.22.60	chart.timeline.label.height .....	4624
6.22.61	chart.timeline.label.textBorderColor .....	4625
6.22.62	chart.timeline.label.textBorderWidth .....	4625
6.22.63	chart.timeline.label.textShadowColor .....	4625
6.22.64	chart.timeline.label.textShadowBlur .....	4626
6.22.65	chart.timeline.label.textShadowOffsetX .....	4626
6.22.66	chart.timeline.label.textShadowOffsetY .....	4627
6.22.67	chart.timeline.label.rich.object .....	4627
6.22.68	chart.timeline.label.emphasis.object .....	4628
6.22.69	chart.timeline.label.emphasis.show .....	4629
6.22.70	chart.timeline.label.emphasis.interval .....	4629
6.22.71	chart.timeline.label.emphasis.rotate .....	4629
6.22.72	chart.timeline.label.emphasis.formatter .....	4630
6.22.73	chart.timeline.label.emphasis.color .....	4630
6.22.74	chart.timeline.label.emphasis.fontStyle .....	4631



	Erlaubte Werte .....	4631
6.22.75	chart.timeline.label.emphasis.fontWeight.....	4631
	Erlaubte Werte .....	4632
6.22.76	chart.timeline.label.emphasis.fontFamily .....	4632
6.22.77	chart.timeline.label.emphasis.fontSize.....	4632
6.22.78	chart.timeline.label.emphasis.align .....	4633
	Erlaubte Werte .....	4633
6.22.79	chart.timeline.label.emphasis.verticalAlign.....	4633
	Erlaubte Werte .....	4634
6.22.80	chart.timeline.label.emphasis.lineHeight .....	4634
6.22.81	chart.timeline.label.emphasis.backgroundColor .....	4634
6.22.82	chart.timeline.label.emphasis.borderColor .....	4635
6.22.83	chart.timeline.label.emphasis.borderWidth .....	4636
6.22.84	chart.timeline.label.emphasis.borderRadius.....	4636
6.22.85	chart.timeline.label.emphasis.padding.....	4636
6.22.86	chart.timeline.label.emphasis.shadowColor .....	4637
6.22.87	chart.timeline.label.emphasis.shadowBlur .....	4637
6.22.88	chart.timeline.label.emphasis.shadowOffsetX .....	4637
6.22.89	chart.timeline.label.emphasis.shadowOffsetY .....	4638
6.22.90	chart.timeline.label.emphasis.width.....	4638
6.22.91	chart.timeline.label.emphasis.height.....	4639
6.22.92	chart.timeline.label.emphasis.textBorderColor .....	4639
6.22.93	chart.timeline.label.emphasis.textBorderWidth .....	4639
6.22.94	chart.timeline.label.emphasis.textShadowColor .....	4640
6.22.95	chart.timeline.label.emphasis.textShadowBlur .....	4640
6.22.96	chart.timeline.label.emphasis.textShadowOffsetX .....	4641
6.22.97	chart.timeline.label.emphasis.textShadowOffsetY .....	4641
6.22.98	chart.timeline.label.emphasis.rich.object .....	4641
6.22.99	chart.timeline.itemStyle.object .....	4642
6.22.100	chart.timeline.itemStyle.color .....	4643
6.22.101	chart.timeline.itemStyle.borderColor .....	4644
6.22.102	chart.timeline.itemStyle.borderWidth .....	4645
6.22.103	chart.timeline.itemStyle.borderType .....	4645
	Erlaubte Werte .....	4645
6.22.104	chart.timeline.itemStyle.shadowBlur.....	4646
6.22.105	chart.timeline.itemStyle.shadowColor.....	4646

6.22.106	chart.timeline.itemStyle.shadowOffsetX .....	4646
6.22.107	chart.timeline.itemStyle.shadowOffsetY .....	4647
6.22.108	chart.timeline.itemStyle.opacity .....	4647
6.22.109	chart.timeline.itemStyle.emphasis.object .....	4647
6.22.110	chart.timeline.itemStyle.emphasis.color .....	4648
6.22.111	chart.timeline.itemStyle.emphasis.borderColor .....	4649
6.22.112	chart.timeline.itemStyle.emphasis.borderWidth .....	4650
6.22.113	chart.timeline.itemStyle.emphasis.borderType .....	4650
	Erlaubte Werte .....	4650
6.22.114	chart.timeline.itemStyle.emphasis.shadowBlur .....	4651
6.22.115	chart.timeline.itemStyle.emphasis.shadowColor .....	4651
6.22.116	chart.timeline.itemStyle.emphasis.shadowOffsetX .....	4651
6.22.117	chart.timeline.itemStyle.emphasis.shadowOffsetY .....	4652
6.22.118	chart.timeline.itemStyle.emphasis.opacity .....	4652
6.22.119	chart.timeline.checkpointStyle.object .....	4652
6.22.120	chart.timeline.checkpointStyle.symbol .....	4653
	Erlaubte Werte .....	4653
6.22.121	chart.timeline.checkpointStyle.symbolSize .....	4654
6.22.122	chart.timeline.checkpointStyle.symbolRotate .....	4654
6.22.123	chart.timeline.checkpointStyle.symbolKeepAspect .....	4654
6.22.124	chart.timeline.checkpointStyle.symbolOffset .....	4655
6.22.125	chart.timeline.checkpointStyle.color .....	4655
6.22.126	chart.timeline.checkpointStyle.borderWidth .....	4655
6.22.127	chart.timeline.checkpointStyle.borderColor .....	4656
6.22.128	chart.timeline.checkpointStyle.animation .....	4656
6.22.129	chart.timeline.checkpointStyle.animationDuration .....	4657
6.22.130	chart.timeline.checkpointStyle.animationEasing .....	4657
	Erlaubte Werte .....	4657
6.22.131	chart.timeline.controlStyle.object .....	4658
6.22.132	chart.timeline.controlStyle.show .....	4658
6.22.133	chart.timeline.controlStyle.showPlayBtn .....	4659
6.22.134	chart.timeline.controlStyle.showPrevBtn .....	4659
6.22.135	chart.timeline.controlStyle.showNextBtn .....	4659
6.22.136	chart.timeline.controlStyle.itemSize .....	4660
6.22.137	chart.timeline.controlStyle.itemGap .....	4660
6.22.138	chart.timeline.controlStyle.position .....	4660

	Erlaubte Werte .....	4661
6.22.139	chart.timeline.controlStyle.playIcon .....	4661
6.22.140	chart.timeline.controlStyle.stopIcon .....	4661
6.22.141	chart.timeline.controlStyle.prevIcon .....	4662
6.22.142	chart.timeline.controlStyle.nextIcon .....	4662
6.22.143	chart.timeline.controlStyle.color .....	4663
6.22.144	chart.timeline.controlStyle.borderColor .....	4663
6.22.145	chart.timeline.controlStyle.borderWidth .....	4664
6.22.146	chart.timeline.controlStyle.emphasis.object .....	4664
6.22.147	chart.timeline.controlStyle.emphasis.color .....	4664
6.22.148	chart.timeline.controlStyle.emphasis.borderColor .....	4665
6.22.149	chart.timeline.controlStyle.emphasis.borderWidth .....	4665
6.23	ECharts (5) Title .....	4665
6.23.1	chart.title.object .....	4665
6.23.2	chart.title.id .....	4666
6.23.3	chart.title.show .....	4666
6.23.4	chart.title.text .....	4667
6.23.5	chart.title.link .....	4667
6.23.6	chart.title.target .....	4667
6.23.7	chart.title.textStyle.object .....	4668
6.23.8	chart.title.textStyle.color .....	4668
6.23.9	chart.title.textStyle.fontStyle .....	4668
	Erlaubte Werte .....	4669
6.23.10	chart.title.textStyle.fontWeight .....	4669
6.23.11	chart.title.textStyle.fontFamily .....	4669
6.23.12	chart.title.textStyle.fontSize .....	4670
6.23.13	chart.title.textStyle.align .....	4670
6.23.14	chart.title.textStyle.verticalAlign .....	4671
6.23.15	chart.title.textStyle.lineHeight .....	4671
6.23.16	chart.title.textStyle.backgroundColor .....	4672
6.23.17	chart.title.textStyle.borderColor .....	4672
6.23.18	chart.title.textStyle.borderWidth .....	4673
6.23.19	chart.title.textStyle.borderRadius .....	4673
6.23.20	chart.title.textStyle.padding .....	4674
6.23.21	chart.title.textStyle.shadowColor .....	4674
6.23.22	chart.title.textStyle.shadowBlur .....	4675

6.23.23	chart.title.textStyle.shadowOffsetX.....	4675
6.23.24	chart.title.textStyle.shadowOffsetY .....	4675
6.23.25	chart.title.textStyle.width .....	4676
	Erlaubte Werte .....	4676
6.23.26	chart.title.textStyle.height .....	4676
6.23.27	chart.title.textStyle.textBorderColor .....	4677
6.23.28	chart.title.textStyle.textBorderWidth .....	4677
6.23.29	chart.title.textStyle.textShadowColor .....	4677
6.23.30	chart.title.textStyle.textShadowBlur .....	4678
6.23.31	chart.title.textStyle.textShadowOffsetX.....	4678
6.23.32	chart.title.textStyle.textShadowOffsetY .....	4678
6.23.33	chart.title.textStyle.rich.....	4679
6.23.34	chart.title.subtext .....	4680
6.23.35	chart.title.sublink.....	4681
6.23.36	chart.title.subtarget.....	4681
6.23.37	chart.title.subtextStyle.object .....	4681
6.23.38	chart.title.subtextStyle.color .....	4682
6.23.39	chart.title.subtextStyle.fontStyle.....	4682
	Erlaubte Werte .....	4682
6.23.40	chart.title.subtextStyle.fontWeight .....	4683
6.23.41	chart.title.subtextStyle.fontFamily .....	4683
6.23.42	chart.title.subtextStyle.fontSize .....	4683
6.23.43	chart.title.subtextStyle.align.....	4684
	Erlaubte Werte .....	4684
6.23.44	chart.title.subtextStyle.verticalAlign .....	4685
6.23.45	chart.title.subtextStyle.lineHeight.....	4685
6.23.46	chart.title.subtextStyle.backgroundColor.....	4686
6.23.47	chart.title.subtextStyle.borderColor.....	4686
6.23.48	chart.title.subtextStyle.borderWidth.....	4687
6.23.49	chart.title.subtextStyle.borderRadius .....	4687
6.23.50	chart.title.subtextStyle.padding .....	4687
6.23.51	chart.title.subtextStyle.shadowColor.....	4688
6.23.52	chart.title.subtextStyle.shadowBlur.....	4688
6.23.53	chart.title.subtextStyle.shadowOffsetX.....	4689
6.23.54	chart.title.subtextStyle.shadowOffsetY .....	4689
6.23.55	chart.title.subtextStyle.width .....	4690

	Erlaubte Werte .....	4690
6.23.56	chart.title.subtextStyle.height .....	4690
6.23.57	chart.title.subtextStyle.textBorderColor .....	4691
6.23.58	chart.title.subtextStyle.textBorderWidth .....	4691
6.23.59	chart.title.subtextStyle.textShadowColor .....	4692
6.23.60	chart.title.subtextStyle.textShadowBlur .....	4692
6.23.61	chart.title.subtextStyle.textShadowOffsetX .....	4692
6.23.62	chart.title.subtextStyle.textShadowOffsetY .....	4693
6.23.63	chart.title.subtextStyle.rich .....	4693
6.23.64	chart.title.triggerEvent .....	4694
6.23.65	chart.title.padding .....	4695
6.23.66	chart.title.itemGap .....	4695
6.23.67	chart.title.zlevel .....	4696
6.23.68	chart.title.z .....	4696
6.23.69	chart.title.left .....	4696
	Erlaubte Werte .....	4697
6.23.70	chart.title.top .....	4697
	Erlaubte Werte .....	4698
6.23.71	chart.title.right .....	4698
	Erlaubte Werte .....	4698
6.23.72	chart.title.bottom .....	4698
	Erlaubte Werte .....	4699
6.23.73	chart.title.backgroundColor .....	4699
6.23.74	chart.title.borderColor .....	4699
6.23.75	chart.title.borderWidth .....	4700
6.23.76	chart.title.borderRadius .....	4700
6.23.77	chart.title.shadowBlur .....	4700
6.23.78	chart.title.shadowColor .....	4701
6.23.79	chart.title.shadowOffsetX .....	4701
6.23.80	chart.title.shadowOffsetY .....	4702
6.23.81	chart.title.textStyle.overflow .....	4702
	Erlaubte Werte .....	4703
6.23.82	chart.title.textStyle.ellipsis .....	4703
6.23.83	chart.title.subtextStyle.overflow .....	4703
	Erlaubte Werte .....	4703
6.23.84	chart.title.subtextStyle.ellipsis .....	4704

6.23.85	chart.title.textAlign .....	4704
	Erlaubte Werte .....	4704
6.23.86	chart.title.textVerticalAlign .....	4704
	Erlaubte Werte .....	4705
<b>6.24</b>	<b>ECharts (5) Toolbox.....</b>	<b>4705</b>
6.24.1	chart.toolbox.id.....	4705
6.24.2	chart.toolbox.show .....	4705
6.24.3	chart.toolbox.orient.....	4706
	Erlaubte Werte .....	4706
6.24.4	chart.toolbox.itemSize .....	4706
6.24.5	chart.toolbox.itemGap .....	4706
6.24.6	chart.toolbox.showTitle .....	4707
6.24.7	chart.toolbox.feature.saveAsImage.....	4707
6.24.8	chart.toolbox.feature.saveAsImage.type .....	4708
6.24.9	chart.toolbox.feature.saveAsImage.name .....	4708
6.24.10	chart.toolbox.feature.saveAsImage.backgroundColor .....	4708
6.24.11	chart.toolbox.feature.saveAsImage.excludeComponents .....	4709
6.24.12	chart.toolbox.feature.saveAsImage.show .....	4709
6.24.13	chart.toolbox.feature.saveAsImage.title .....	4709
6.24.14	chart.toolbox.feature.saveAsImage.icon.....	4710
6.24.15	chart.toolbox.feature.saveAsImage.iconStyle .....	4710
6.24.16	chart.toolbox.feature.saveAsImage.iconStyle.color.....	4710
6.24.17	chart.toolbox.feature.saveAsImage.iconStyle.borderColor.....	4711
6.24.18	chart.toolbox.feature.saveAsImage.iconStyle.borderWidth.....	4711
6.24.19	chart.toolbox.feature.saveAsImage.iconStyle.borderType.....	4712
	Erlaubte Werte .....	4712
6.24.20	chart.toolbox.feature.saveAsImage.iconStyle.shadowBlur .....	4712
6.24.21	chart.toolbox.feature.saveAsImage.iconStyle.shadowColor .....	4712
6.24.22	chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetX .....	4713
6.24.23	chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetY .....	4713
6.24.24	chart.toolbox.feature.saveAsImage.iconStyle.opacity.....	4713
6.24.25	chart.toolbox.feature.saveAsImage.iconStyle.textPosition.....	4714
	Erlaubte Werte .....	4714
6.24.26	chart.toolbox.feature.saveAsImage.iconStyle.textAlign .....	4714
	Erlaubte Werte .....	4715
6.24.27	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.color.....	4715

6.24.28	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderColor	4715
6.24.29	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderWidth	4715
6.24.30	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderType	4716
	Erlaubte Werte	4716
6.24.31	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowBlur	4716
6.24.32	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowColor	4717
6.24.33	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetX	4717
6.24.34	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetY	4717
6.24.35	chart.toolbox.feature.saveAsImage.iconStyle.emphasis.opacity	4718
6.24.36	chart.toolbox.feature.saveAsImage.pixelRatio	4718
6.24.37	chart.toolbox.feature.restore	4718
6.24.38	chart.toolbox.feature.restore.show	4719
6.24.39	chart.toolbox.feature.restore.title	4719
6.24.40	chart.toolbox.feature.restore.icon	4719
6.24.41	chart.toolbox.feature.restore.iconStyle	4720
6.24.42	chart.toolbox.feature.restore.iconStyle.color	4720
6.24.43	chart.toolbox.feature.restore.iconStyle.borderColor	4720
6.24.44	chart.toolbox.feature.restore.iconStyle.borderWidth	4721
6.24.45	chart.toolbox.feature.restore.iconStyle.borderType	4721
	Erlaubte Werte	4721
6.24.46	chart.toolbox.feature.restore.iconStyle.shadowBlur	4722
6.24.47	chart.toolbox.feature.restore.iconStyle.shadowColor	4722
6.24.48	chart.toolbox.feature.restore.iconStyle.shadowOffsetX	4722
6.24.49	chart.toolbox.feature.restore.iconStyle.shadowOffsetY	4723
6.24.50	chart.toolbox.feature.restore.iconStyle.opacity	4723
6.24.51	chart.toolbox.feature.restore.iconStyle.textPosition	4723
	Erlaubte Werte	4724
6.24.52	chart.toolbox.feature.restore.iconStyle.textAlign	4724
	Erlaubte Werte	4724
6.24.53	chart.toolbox.feature.restore.iconStyle.emphasis.color	4724
6.24.54	chart.toolbox.feature.restore.iconStyle.emphasis.borderColor	4725
6.24.55	chart.toolbox.feature.restore.iconStyle.emphasis.borderWidth	4725
6.24.56	chart.toolbox.feature.restore.iconStyle.emphasis.borderType	4725
	Erlaubte Werte	4726
6.24.57	chart.toolbox.feature.restore.iconStyle.emphasis.shadowBlur	4726
6.24.58	chart.toolbox.feature.restore.iconStyle.emphasis.shadowColor	4726

6.24.59	chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetX.....	4727
6.24.60	chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetY.....	4727
6.24.61	chart.toolbox.feature.restore.iconStyle.emphasis.opacity.....	4727
6.24.62	chart.toolbox.feature.dataView.....	4728
6.24.63	chart.toolbox.feature.dataView.show.....	4728
6.24.64	chart.toolbox.feature.dataView.title.....	4728
6.24.65	chart.toolbox.feature.dataView.icon.....	4729
6.24.66	chart.toolbox.feature.dataView.iconStyle.....	4729
6.24.67	chart.toolbox.feature.dataView.iconStyle.color.....	4729
6.24.68	chart.toolbox.feature.dataView.iconStyle.borderColor.....	4730
6.24.69	chart.toolbox.feature.dataView.iconStyle.borderWidth.....	4730
6.24.70	chart.toolbox.feature.dataView.iconStyle.borderType.....	4731
	Erlaubte Werte.....	4731
6.24.71	chart.toolbox.feature.dataView.iconStyle.shadowBlur.....	4731
6.24.72	chart.toolbox.feature.dataView.iconStyle.shadowColor.....	4731
6.24.73	chart.toolbox.feature.dataView.iconStyle.shadowOffsetX.....	4732
6.24.74	chart.toolbox.feature.dataView.iconStyle.shadowOffsetY.....	4732
6.24.75	chart.toolbox.feature.dataView.iconStyle.opacity.....	4732
6.24.76	chart.toolbox.feature.dataView.iconStyle.textPosition.....	4733
	Erlaubte Werte.....	4733
6.24.77	chart.toolbox.feature.dataView.iconStyle.textAlign.....	4733
	Erlaubte Werte.....	4734
6.24.78	chart.toolbox.feature.dataView.iconStyle.emphasis.....	4734
6.24.79	chart.toolbox.feature.dataView.iconStyle.emphasis.color.....	4734
6.24.80	chart.toolbox.feature.dataView.iconStyle.emphasis.borderColor.....	4734
6.24.81	chart.toolbox.feature.dataView.iconStyle.emphasis.borderWidth.....	4735
6.24.82	chart.toolbox.feature.dataView.iconStyle.emphasis.borderType.....	4735
	Erlaubte Werte.....	4735
6.24.83	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowBlur.....	4736
6.24.84	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowColor.....	4736
6.24.85	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetX.....	4736
6.24.86	chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetY.....	4737
6.24.87	chart.toolbox.feature.dataView.iconStyle.emphasis.opacity.....	4737
6.24.88	chart.toolbox.feature.dataView.readOnly.....	4737
6.24.89	chart.toolbox.feature.dataView.optionToContent.....	4738
6.24.90	chart.toolbox.feature.dataView.contentToOption.....	4738



6.24.91	chart.toolbox.feature.dataView.lang.....	4739
6.24.92	chart.toolbox.feature.dataView.backgroundColor.....	4739
6.24.93	chart.toolbox.feature.dataView.textareaColor.....	4740
6.24.94	chart.toolbox.feature.dataView.textareaBorderColor.....	4740
6.24.95	chart.toolbox.feature.dataView.textColor.....	4740
6.24.96	chart.toolbox.feature.dataView.buttonColor.....	4741
6.24.97	chart.toolbox.feature.dataView.buttonTextColor.....	4741
6.24.98	chart.toolbox.feature.dataZoom.....	4741
6.24.99	chart.toolbox.feature.dataZoom.show.....	4742
6.24.100	chart.toolbox.feature.dataZoom.title.....	4742
6.24.101	chart.toolbox.feature.dataZoom.title.zoom.....	4742
6.24.102	chart.toolbox.feature.dataZoom.title.back.....	4743
6.24.103	chart.toolbox.feature.dataZoom.icon.....	4743
6.24.104	chart.toolbox.feature.dataZoom.icon.zoom.....	4743
6.24.105	chart.toolbox.feature.dataZoom.icon.back.....	4744
6.24.106	chart.toolbox.feature.dataZoom.iconStyle.....	4744
6.24.107	chart.toolbox.feature.dataZoom.iconStyle.color.....	4745
6.24.108	chart.toolbox.feature.dataZoom.iconStyle.borderColor.....	4745
6.24.109	chart.toolbox.feature.dataZoom.iconStyle.borderWidth.....	4745
6.24.110	chart.toolbox.feature.dataZoom.iconStyle.borderType.....	4746
	Erlaubte Werte.....	4746
6.24.111	chart.toolbox.feature.dataZoom.iconStyle.shadowBlur.....	4746
6.24.112	chart.toolbox.feature.dataZoom.iconStyle.shadowColor.....	4746
6.24.113	chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetX.....	4747
6.24.114	chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetY.....	4747
6.24.115	chart.toolbox.feature.dataZoom.iconStyle.opacity.....	4748
6.24.116	chart.toolbox.feature.dataZoom.iconStyle.textPosition.....	4748
	Erlaubte Werte.....	4748
6.24.117	chart.toolbox.feature.dataZoom.iconStyle.textAlign.....	4748
	Erlaubte Werte.....	4749
6.24.118	chart.toolbox.feature.dataZoom.iconStyle.emphasis.color.....	4749
6.24.119	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderColor.....	4749
6.24.120	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderWidth.....	4750
6.24.121	chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderType.....	4750
	Erlaubte Werte.....	4750
6.24.122	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowBlur.....	4750

6.24.123	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowColor .....	4751
6.24.124	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetX .....	4751
6.24.125	chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetY .....	4751
6.24.126	chart.toolbox.feature.dataZoom.iconStyle.emphasis.opacity .....	4752
6.24.127	chart.toolbox.feature.dataZoom.xAxisIndex .....	4752
6.24.128	chart.toolbox.feature.dataZoom.yAxisIndex .....	4752
6.24.129	chart.toolbox.feature.magicType .....	4753
6.24.130	chart.toolbox.feature.magicType.show .....	4753
6.24.131	chart.toolbox.feature.magicType.type .....	4754
6.24.132	chart.toolbox.feature.magicType.title .....	4754
6.24.133	chart.toolbox.feature.magicType.title.line .....	4754
6.24.134	chart.toolbox.feature.magicType.title.bar .....	4755
6.24.135	chart.toolbox.feature.magicType.title.stack .....	4755
6.24.136	chart.toolbox.feature.magicType.title.tiled .....	4755
6.24.137	chart.toolbox.feature.magicType.icon .....	4756
6.24.138	chart.toolbox.feature.magicType.icon.line .....	4756
6.24.139	chart.toolbox.feature.magicType.icon.bar .....	4756
6.24.140	chart.toolbox.feature.magicType.icon.stack .....	4757
6.24.141	chart.toolbox.feature.magicType.icon.tiled .....	4757
6.24.142	chart.toolbox.feature.magicType.iconStyle .....	4758
6.24.143	chart.toolbox.feature.magicType.iconStyle.color .....	4758
6.24.144	chart.toolbox.feature.magicType.iconStyle.borderColor .....	4759
6.24.145	chart.toolbox.feature.magicType.iconStyle.borderWidth .....	4759
6.24.146	chart.toolbox.feature.magicType.iconStyle.borderType .....	4759
	Erlaubte Werte .....	4760
6.24.147	chart.toolbox.feature.magicType.iconStyle.shadowBlur .....	4760
6.24.148	chart.toolbox.feature.magicType.iconStyle.shadowColor .....	4760
6.24.149	chart.toolbox.feature.magicType.iconStyle.shadowOffsetX .....	4760
6.24.150	chart.toolbox.feature.magicType.iconStyle.shadowOffsetY .....	4761
6.24.151	chart.toolbox.feature.magicType.iconStyle.opacity .....	4761
6.24.152	chart.toolbox.feature.magicType.iconStyle.textPosition .....	4761
	Erlaubte Werte .....	4762
6.24.153	chart.toolbox.feature.magicType.iconStyle.textAlign .....	4762
	Erlaubte Werte .....	4762
6.24.154	chart.toolbox.feature.magicType.iconStyle.emphasis .....	4762
6.24.155	chart.toolbox.feature.magicType.iconStyle.emphasis.color .....	4763

6.24.156	chart.toolbox.feature.magicType.iconStyle.emphasis.borderColor .....	4763
6.24.157	chart.toolbox.feature.magicType.iconStyle.emphasis.borderWidth .....	4763
6.24.158	chart.toolbox.feature.magicType.iconStyle.emphasis.borderType .....	4764
	Erlaubte Werte .....	4764
6.24.159	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowBlur .....	4764
6.24.160	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowColor .....	4765
6.24.161	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetX .....	4765
6.24.162	chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetY .....	4765
6.24.163	chart.toolbox.feature.magicType.iconStyle.emphasis.opacity .....	4766
6.24.164	chart.toolbox.feature.magicType.option .....	4766
6.24.165	chart.toolbox.feature.magicType.option.line .....	4766
6.24.166	chart.toolbox.feature.magicType.option.bar .....	4767
6.24.167	chart.toolbox.feature.magicType.option.stack .....	4767
6.24.168	chart.toolbox.feature.magicType.option.tiled.....	4767
6.24.169	chart.toolbox.feature.magicType.seriesIndex .....	4768
6.24.170	chart.toolbox.feature.magicType.seriesIndex.line .....	4768
6.24.171	chart.toolbox.feature.magicType.seriesIndex.bar.....	4768
6.24.172	chart.toolbox.feature.magicType.seriesIndex.stack .....	4769
6.24.173	chart.toolbox.feature.magicType.seriesIndex.tiled.....	4769
6.24.174	chart.toolbox.feature.brush .....	4769
6.24.175	chart.toolbox.feature.brush.show .....	4770
6.24.176	chart.toolbox.feature.brush.type.....	4770
6.24.177	chart.toolbox.feature.brush.icon.....	4770
6.24.178	chart.toolbox.feature.brush.icon.rect .....	4771
6.24.179	chart.toolbox.feature.brush.icon.polygon .....	4771
6.24.180	chart.toolbox.feature.brush.icon.lineX.....	4772
6.24.181	chart.toolbox.feature.brush.icon.lineY.....	4772
6.24.182	chart.toolbox.feature.brush.icon.keep.....	4773
6.24.183	chart.toolbox.feature.brush.icon.clear.....	4773
6.24.184	chart.toolbox.feature.brush.title .....	4773
6.24.185	chart.toolbox.feature.brush.title.rect.....	4774
6.24.186	chart.toolbox.feature.brush.title.polygon.....	4774
6.24.187	chart.toolbox.feature.brush.title.lineX .....	4774
6.24.188	chart.toolbox.feature.brush.title.lineY .....	4775
6.24.189	chart.toolbox.feature.brush.title.keep .....	4775
6.24.190	chart.toolbox.feature.brush.title.clear .....	4775

6.24.191	chart.toolbox.iconStyle .....	4776
6.24.192	chart.toolbox.iconStyle.color.....	4776
6.24.193	chart.toolbox.iconStyle.borderColor.....	4777
6.24.194	chart.toolbox.iconStyle.borderWidth.....	4777
6.24.195	chart.toolbox.iconStyle.borderType.....	4777
	Erlaubte Werte .....	4778
6.24.196	chart.toolbox.iconStyle.shadowBlur .....	4778
6.24.197	chart.toolbox.iconStyle.shadowColor .....	4778
6.24.198	chart.toolbox.iconStyle.shadowOffsetX .....	4778
6.24.199	chart.toolbox.iconStyle.shadowOffsetY .....	4779
6.24.200	chart.toolbox.iconStyle.opacity.....	4779
6.24.201	chart.toolbox.iconStyle.textPosition .....	4779
	Erlaubte Werte .....	4780
6.24.202	chart.toolbox.iconStyle.textAlign .....	4780
	Erlaubte Werte .....	4780
6.24.203	chart.toolbox.iconStyle.emphasis .....	4780
6.24.204	chart.toolbox.iconStyle.emphasis.color .....	4781
6.24.205	chart.toolbox.iconStyle.emphasis.borderColor.....	4781
6.24.206	chart.toolbox.iconStyle.emphasis.borderWidth.....	4781
6.24.207	chart.toolbox.iconStyle.emphasis.borderType .....	4782
	Erlaubte Werte .....	4782
6.24.208	chart.toolbox.iconStyle.emphasis.shadowBlur .....	4782
6.24.209	chart.toolbox.iconStyle.emphasis.shadowColor .....	4783
6.24.210	chart.toolbox.iconStyle.emphasis.shadowOffsetX.....	4783
6.24.211	chart.toolbox.iconStyle.emphasis.shadowOffsetY .....	4783
6.24.212	chart.toolbox.iconStyle.emphasis.opacity.....	4784
6.24.213	chart.toolbox.zlevel .....	4784
6.24.214	chart.toolbox.z .....	4784
6.24.215	chart.toolbox.left .....	4785
	Erlaubte Werte .....	4785
6.24.216	chart.toolbox.top .....	4785
	Erlaubte Werte .....	4786
6.24.217	chart.toolbox.right.....	4786
6.24.218	chart.toolbox.bottom .....	4786
6.24.219	chart.toolbox.width .....	4787
6.24.220	chart.toolbox.height .....	4787

6.24.221	chart.toolbox.defaulticons .....	4788
6.24.222	chart.toolbox.feature.myToolToggleSeriesVisibility.show .....	4788
6.24.223	chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.show .....	4788
6.24.224	chart.toolbox.feature.RescaleYAxis.function .....	4789
6.24.225	chart.toolbox.feature.myToolRestore.show .....	4789
6.24.226	chart.toolbox.feature.myToolBarChart.show .....	4790
6.24.227	chart.toolbox.feature.myToolLineChart.show .....	4790
6.24.228	chart.toolbox.feature.myToolStackedChart.show .....	4790
6.24.229	chart.toolbox.feature.myToolRestore.additionalscript .....	4791
6.24.230	chart.toolbox.feature.myToolBarChart.additionalscript .....	4791
6.24.231	chart.toolbox.feature.myToolLineChart.additionalscript .....	4792
6.24.232	chart.toolbox.feature.myToolStackedChart.additionalscript .....	4792
6.24.233	chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript .....	4793
6.24.234	chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.additionalscript .....	4793
6.25	ECharts (5) Tooltip .....	4794
6.25.1	chart.tooltip.object .....	4794
6.25.2	chart.tooltip.show .....	4794
6.25.3	chart.tooltip.trigger .....	4795
	Erlaubte Werte .....	4795
6.25.4	chart.tooltip.axisPointer.object .....	4795
6.25.5	chart.tooltip.axisPointer.type .....	4796
	Erlaubte Werte .....	4796
6.25.6	chart.tooltip.axisPointer.axis .....	4796
	Erlaubte Werte .....	4797
6.25.7	chart.tooltip.axisPointer.snap .....	4797
6.25.8	chart.tooltip.axisPointer.z .....	4797
6.25.9	chart.tooltip.axisPointer.label.object .....	4798
6.25.10	chart.tooltip.axisPointer.label.show .....	4798
6.25.11	chart.tooltip.axisPointer.label.precision .....	4798
6.25.12	chart.tooltip.axisPointer.label.formatter .....	4799
6.25.13	chart.tooltip.axisPointer.label.margin .....	4800
6.25.14	chart.tooltip.axisPointer.label.color .....	4801
6.25.15	chart.tooltip.axisPointer.label.fontStyle .....	4801
	Erlaubte Werte .....	4801
6.25.16	chart.tooltip.axisPointer.label.fontWeight .....	4802
	Erlaubte Werte .....	4802

6.25.17	chart.tooltip.axisPointer.label.fontFamily .....	4802
6.25.18	chart.tooltip.axisPointer.label.fontSize.....	4803
6.25.19	chart.tooltip.axisPointer.label.lineHeight.....	4803
6.25.20	chart.tooltip.axisPointer.label.width.....	4804
6.25.21	chart.tooltip.axisPointer.label.height.....	4804
6.25.22	chart.tooltip.axisPointer.label.textBorderColor .....	4804
6.25.23	chart.tooltip.axisPointer.label.textBorderWidth .....	4805
6.25.24	chart.tooltip.axisPointer.label.textShadowColor .....	4805
6.25.25	chart.tooltip.axisPointer.label.textShadowBlur .....	4806
6.25.26	chart.tooltip.axisPointer.label.textShadowOffsetX.....	4806
6.25.27	chart.tooltip.axisPointer.label.textShadowOffsetY .....	4806
6.25.28	chart.tooltip.axisPointer.label.padding .....	4807
6.25.29	chart.tooltip.axisPointer.label.backgroundColor .....	4807
6.25.30	chart.tooltip.axisPointer.label.borderColor.....	4808
6.25.31	chart.tooltip.axisPointer.label.borderWidth .....	4808
6.25.32	chart.tooltip.axisPointer.label.shadowBlur .....	4808
6.25.33	chart.tooltip.axisPointer.label.shadowColor .....	4809
6.25.34	chart.tooltip.axisPointer.label.shadowOffsetX.....	4809
6.25.35	chart.tooltip.axisPointer.label.shadowOffsetY .....	4810
6.25.36	chart.tooltip.axisPointer.lineStyle.object .....	4810
6.25.37	chart.tooltip.axisPointer.lineStyle.color .....	4810
6.25.38	chart.tooltip.axisPointer.lineStyle.width .....	4812
6.25.39	chart.tooltip.axisPointer.lineStyle.type .....	4812
	Erlaubte Werte .....	4812
6.25.40	chart.tooltip.axisPointer.lineStyle.shadowBlur.....	4812
6.25.41	chart.tooltip.axisPointer.lineStyle.shadowColor.....	4813
6.25.42	chart.tooltip.axisPointer.lineStyle.shadowOffsetX.....	4813
6.25.43	chart.tooltip.axisPointer.lineStyle.shadowOffsetY.....	4814
6.25.44	chart.tooltip.axisPointer.lineStyle.opacity .....	4814
6.25.45	chart.tooltip.axisPointer.shadowStyle.object .....	4814
6.25.46	chart.tooltip.axisPointer.shadowStyle.color .....	4815
6.25.47	chart.tooltip.axisPointer.shadowStyle.shadowBlur.....	4816
6.25.48	chart.tooltip.axisPointer.shadowStyle.shadowColor.....	4817
6.25.49	chart.tooltip.axisPointer.shadowStyle.shadowOffsetX.....	4817
6.25.50	chart.tooltip.axisPointer.shadowStyle.shadowOffsetY.....	4818
6.25.51	chart.tooltip.axisPointer.shadowStyle.opacity .....	4818

6.25.52	chart.tooltip.axisPointer.crossStyle.object.....	4818
6.25.53	chart.tooltip.axisPointer.crossStyle.color.....	4819
6.25.54	chart.tooltip.axisPointer.crossStyle.width.....	4820
6.25.55	chart.tooltip.axisPointer.crossStyle.type .....	4821
	Erlaubte Werte .....	4821
6.25.56	chart.tooltip.axisPointer.crossStyle.shadowBlur .....	4821
6.25.57	chart.tooltip.axisPointer.crossStyle.shadowColor .....	4822
6.25.58	chart.tooltip.axisPointer.crossStyle.shadowOffsetX .....	4822
6.25.59	chart.tooltip.axisPointer.crossStyle.shadowOffsetY .....	4822
6.25.60	chart.tooltip.axisPointer.crossStyle.opacity .....	4823
6.25.61	chart.tooltip.showContent.....	4823
6.25.62	chart.tooltip.alwaysShowContent.....	4823
6.25.63	chart.tooltip.triggerOn .....	4824
	Erlaubte Werte .....	4824
6.25.64	chart.tooltip.showDelay .....	4825
6.25.65	chart.tooltip.hideDelay.....	4825
6.25.66	chart.tooltip.enterable .....	4825
6.25.67	chart.tooltip.renderMode.....	4826
	Erlaubte Werte .....	4826
6.25.68	chart.tooltip.confine .....	4826
6.25.69	chart.tooltip.transitionDuration .....	4827
6.25.70	chart.tooltip.position.....	4827
6.25.71	chart.tooltip.formatter .....	4828
6.25.72	chart.tooltip.backgroundColor .....	4831
6.25.73	chart.tooltip.borderColor.....	4831
6.25.74	chart.tooltip.borderWidth .....	4832
6.25.75	chart.tooltip.padding .....	4832
6.25.76	chart.tooltip.textStyle.object.....	4833
6.25.77	chart.tooltip.textStyle.color .....	4833
6.25.78	chart.tooltip.textStyle.fontStyle .....	4833
	Erlaubte Werte .....	4834
6.25.79	chart.tooltip.textStyle.fontWeight.....	4834
	Erlaubte Werte .....	4834
6.25.80	chart.tooltip.textStyle.fontFamily .....	4834
6.25.81	chart.tooltip.textStyle.fontSize.....	4835
6.25.82	chart.tooltip.textStyle.lineHeight .....	4835

6.25.83	chart.tooltip.textStyle.width .....	4836
6.25.84	chart.tooltip.textStyle.height .....	4836
6.25.85	chart.tooltip.textStyle.textBorderColor .....	4837
6.25.86	chart.tooltip.textStyle.textBorderWidth .....	4837
6.25.87	chart.tooltip.textStyle.textShadowColor .....	4837
6.25.88	chart.tooltip.textStyle.textShadowBlur .....	4838
6.25.89	chart.tooltip.textStyle.textShadowOffsetX .....	4838
6.25.90	chart.tooltip.textStyle.textShadowOffsetY .....	4838
6.25.91	chart.tooltip.extraCssText .....	4839
6.25.92	chart.tooltip.axisPointer.label.textBorderStyle .....	4839
	Erlaubte Werte .....	4840
6.25.93	chart.tooltip.axisPointer.label.textBorderDashOffset .....	4840
6.25.94	chart.tooltip.axisPointer.label.overflow .....	4840
	Erlaubte Werte .....	4841
6.25.95	chart.tooltip.axisPointer.label.ellipsis .....	4841
6.25.96	chart.tooltip.axisPointer.lineStyle.dashOffset .....	4841
6.25.97	chart.tooltip.axisPointer.lineStyle.cap .....	4841
	Erlaubte Werte .....	4842
6.25.98	chart.tooltip.axisPointer.lineStyle.join .....	4842
	Erlaubte Werte .....	4843
6.25.99	chart.tooltip.axisPointer.lineStyle.miterLimit .....	4843
6.25.100	chart.tooltip.axisPointer.crossStyle.miterLimit .....	4843
6.25.101	chart.tooltip.axisPointer.crossStyle.join .....	4843
	Erlaubte Werte .....	4844
6.25.102	chart.tooltip.axisPointer.crossStyle.cap .....	4844
	Erlaubte Werte .....	4845
6.25.103	chart.tooltip.axisPointer.crossStyle.dashOffset .....	4845
6.26	<b>ECharts (5) Visual Map .....</b>	<b>4845</b>
6.26.1	chart.visualMap.object .....	4845
6.26.2	chart.visualMap.type .....	4847
	Erlaubte Werte .....	4847
6.26.3	chart.visualMap.id .....	4847
6.26.4	chart.visualMap.min .....	4847
6.26.5	chart.visualMap.max .....	4848
6.26.6	chart.visualMap.range .....	4848
6.26.7	chart.visualMap.calculable .....	4849



6.26.8	chart.visualMap.realtime.....	4850
6.26.9	chart.visualMap.inverse.....	4850
6.26.10	chart.visualMap.precision .....	4851
6.26.11	chart.visualMap.itemWidth .....	4851
6.26.12	chart.visualMap.itemHeight.....	4852
6.26.13	chart.visualMap.align.....	4852
	Erlaubte Werte .....	4853
6.26.14	chart.visualMap.text .....	4853
6.26.15	chart.visualMap.textGap.....	4853
6.26.16	chart.visualMap.show .....	4854
6.26.17	chart.visualMap.dimension .....	4854
6.26.18	chart.visualMap.seriesIndex.....	4855
6.26.19	chart.visualMap.hoverLink .....	4855
6.26.20	chart.visualMap.inRange .....	4855
6.26.21	chart.visualMap.outOfRange.....	4860
6.26.22	chart.visualMap.controller.object.....	4865
6.26.23	chart.visualMap.controller.inRange .....	4866
6.26.24	chart.visualMap.controller.outOfRange .....	4870
6.26.25	chart.visualMap.zlevel .....	4875
6.26.26	chart.visualMap.z .....	4876
6.26.27	chart.visualMap.left .....	4876
	Erlaubte Werte .....	4877
6.26.28	chart.visualMap.top .....	4877
	Erlaubte Werte .....	4877
6.26.29	chart.visualMap.right.....	4877
	Erlaubte Werte .....	4878
6.26.30	chart.visualMap.bottom .....	4878
	Erlaubte Werte .....	4878
6.26.31	chart.visualMap.orient.....	4878
	Erlaubte Werte .....	4879
6.26.32	chart.visualMap.padding.....	4879
6.26.33	chart.visualMap.backgroundColor .....	4879
6.26.34	chart.visualMap.borderColor .....	4880
6.26.35	chart.visualMap.borderWidth .....	4880
6.26.36	chart.visualMap.color .....	4881
6.26.37	chart.visualMap.textStyle.object .....	4881

6.26.38	chart.visualMap.textStyle.color .....	4881
6.26.39	chart.visualMap.textStyle.fontStyle.....	4882
	Erlaubte Werte .....	4882
6.26.40	chart.visualMap.textStyle.fontWeight .....	4882
	Erlaubte Werte .....	4883
6.26.41	chart.visualMap.textStyle.fontFamily.....	4883
6.26.42	chart.visualMap.textStyle.fontSize .....	4883
6.26.43	chart.visualMap.textStyle.lineHeight.....	4884
6.26.44	chart.visualMap.textStyle.width .....	4884
6.26.45	chart.visualMap.textStyle.height .....	4885
6.26.46	chart.visualMap.textStyle.textBorderColor.....	4885
6.26.47	chart.visualMap.textStyle.textBorderWidth.....	4885
6.26.48	chart.visualMap.textStyle.textShadowColor.....	4886
6.26.49	chart.visualMap.textStyle.textShadowBlur.....	4886
6.26.50	chart.visualMap.textStyle.textShadowOffsetX.....	4886
6.26.51	chart.visualMap.textStyle.textShadowOffsetY.....	4887
6.26.52	chart.visualMap.formatter.....	4887
6.26.53	chart.visualMap.splitNumber.....	4888
6.26.54	chart.visualMap.pieces .....	4888
6.26.55	chart.visualMap.categories .....	4889
6.26.56	chart.visualMap.minOpen .....	4890
6.26.57	chart.visualMap.maxOpen.....	4890
6.26.58	chart.visualMap.selectedMode .....	4890
	Erlaubte Werte .....	4891
6.26.59	chart.visualMap.showLabel.....	4891
6.26.60	chart.visualMap.textStyle.textBorderStyle.....	4891
	Erlaubte Werte .....	4892
6.26.61	chart.visualMap.textStyle.textBorderDashOffset .....	4892
6.26.62	chart.visualMap.textStyle.overflow .....	4892
	Erlaubte Werte .....	4893
6.26.63	chart.visualMap.textStyle.ellipsis .....	4893
6.26.64	chart.visualMap.handleIcon.....	4893
6.26.65	chart.visualMap.handleSize .....	4894
	Erlaubte Werte .....	4894
6.26.66	chart.visualMap.handleStyle.object .....	4894
6.26.67	chart.visualMap.handleStyle.color .....	4894

6.26.68	chart.visualMap.handleStyle.borderColor .....	4895
6.26.69	chart.visualMap.handleStyle.borderWidth .....	4895
6.26.70	chart.visualMap.handleStyle.borderType .....	4895
	Erlaubte Werte .....	4896
6.26.71	chart.visualMap.handleStyle.borderDashOffset .....	4896
6.26.72	chart.visualMap.handleStyle.borderCap .....	4896
	Erlaubte Werte .....	4897
6.26.73	chart.visualMap.handleStyle.borderJoin .....	4897
	Erlaubte Werte .....	4897
6.26.74	chart.visualMap.handleStyle.borderMiterLimit .....	4898
6.26.75	chart.visualMap.handleStyle.shadowBlur.....	4898
6.26.76	chart.visualMap.handleStyle.shadowColor .....	4898
6.26.77	chart.visualMap.handleStyle.shadowOffsetX .....	4899
6.26.78	chart.visualMap.handleStyle.shadowOffsetY.....	4899
6.26.79	chart.visualMap.handleStyle.opacity .....	4900
6.26.80	chart.visualMap.indicatorIcon .....	4900
6.26.81	chart.visualMap.indicatorSize.....	4900
	Erlaubte Werte .....	4901
6.26.82	chart.visualMap.indicatorStyle.object.....	4901
6.26.83	chart.visualMap.indicatorStyle.color.....	4901
6.26.84	chart.visualMap.indicatorStyle.borderColor.....	4901
6.26.85	chart.visualMap.indicatorStyle.borderWidth.....	4902
6.26.86	chart.visualMap.indicatorStyle.borderType.....	4902
	Erlaubte Werte .....	4902
6.26.87	chart.visualMap.indicatorStyle.borderDashOffset .....	4903
6.26.88	chart.visualMap.indicatorStyle.borderCap .....	4903
	Erlaubte Werte .....	4903
6.26.89	chart.visualMap.indicatorStyle.borderJoin .....	4903
	Erlaubte Werte .....	4904
6.26.90	chart.visualMap.indicatorStyle.borderMiterLimit.....	4904
6.26.91	chart.visualMap.indicatorStyle.shadowBlur .....	4905
6.26.92	chart.visualMap.indicatorStyle.shadowColor .....	4905
6.26.93	chart.visualMap.indicatorStyle.shadowOffsetX.....	4905
6.26.94	chart.visualMap.indicatorStyle.shadowOffsetY .....	4906
6.26.95	chart.visualMap.indicatorStyle.opacity.....	4906

<b>7</b>	<b>Leaflet [Options]</b> .....	<b>4907</b>
7.1	Leaflet Buttons .....	4907
7.1.1	leaflet.distancebutton.values .....	4907
7.2	Leaflet General .....	4907
7.2.1	leaflet.height .....	4907
7.3	Leaflet Infobox .....	4908
7.3.1	leaflet.infobox.style .....	4908
7.3.2	leaflet.infobox.style.default .....	4909
7.3.3	leaflet.infobox.html.active .....	4909
7.3.4	leaflet.infobox.html.default .....	4909
7.4	Leaflet Layer .....	4910
7.4.1	leaflet.layer.circle.show .....	4910
7.4.2	leaflet.layer.circle.radius .....	4910
7.4.3	leaflet.layer.circle.color .....	4911
7.4.4	leaflet.layer.circle.fillcolor .....	4911
7.4.5	leaflet.layer.circle.opacity .....	4912
7.4.6	leaflet.layer.circle.weight .....	4912
7.5	Leaflet Map .....	4913
7.5.1	leaflet.map.filename .....	4913
7.5.2	leaflet.map.autoload .....	4913
7.5.3	leaflet.map.tileLayer .....	4914
7.5.4	leaflet.map.viewpoint .....	4914
7.6	Leaflet Marker .....	4915
7.6.1	leaflet.marker.options.title .....	4915
7.6.2	leaflet.marker.event.mouseover .....	4915
7.6.3	leaflet.marker.event.mouseout .....	4916
7.6.4	leaflet.marker.icon.active .....	4916
7.6.5	leaflet.marker.icon.default .....	4916
7.6.6	leaflet.marker.icon.[] .iconUrl .....	4917
7.6.7	leaflet.marker.icon.[] .iconSize .....	4918
7.6.8	leaflet.marker.icon.[] .iconAnchor .....	4919
7.6.9	leaflet.marker.icon.[] .popupAnchor .....	4919
7.6.10	leaflet.marker.icon.[] .shadowUrl .....	4920
7.6.11	leaflet.marker.icon.[] .shadowSize .....	4920
7.6.12	leaflet.marker.icon.[] .shadowAnchor .....	4920

7.6.13	leaflet.marker.icon.active2.....	4921
7.6.14	leaflet.marker.event.click.....	4921
7.6.15	leaflet.markers.autoload.....	4922
<b>8</b>	<b>OrgChart [Options].....</b>	<b>4923</b>
8.1	OrgChart General .....	4923
8.1.1	orgchart.all.options .....	4923
<b>9</b>	<b>PivotTable [Options].....</b>	<b>4924</b>
9.1	PivotTable General.....	4924
9.1.1	pivot.custom.script.before .....	4924
9.1.2	pivot.display.type .....	4924
9.1.3	pivot.all.options .....	4925
9.1.4	pivot.rows.....	4925
9.1.5	pivot.cols .....	4925
9.1.6	pivot.aggregator .....	4926
9.1.7	pivot.aggregatorName .....	4926
9.1.8	pivot.renderer .....	4926
9.1.9	pivot.rowOrder.....	4927
9.1.10	pivot.colOrder .....	4927
9.1.11	pivot.derivedAttributes .....	4927
9.1.12	pivot.dataClass.....	4928
9.1.13	pivot.filter.....	4928
9.1.14	pivot.sorters .....	4929
9.1.15	pivot.rendererOptions .....	4929
9.1.16	pivot.localeStrings.....	4929
9.1.17	pivot.vals .....	4930
9.1.18	pivot.aggregators.....	4930
9.1.19	pivot.renderers.....	4930
9.1.20	pivot.inclusions.....	4931
9.1.21	pivot.exclusions .....	4931
9.1.22	pivot.hiddenAttributes .....	4932
9.1.23	pivot.hiddenFromAggregators.....	4932
9.1.24	pivot.hiddenFromDragDrop .....	4932
9.1.25	pivot.onRefresh.....	4933
9.1.26	pivot.menuLimit.....	4933
9.1.27	pivot.autoSortUnusedAttrs .....	4933

9.1.28	<code>pivot.unusedAttrsVertical</code> .....	4934
9.1.29	<code>pivot.showUI</code> .....	4934
9.1.30	<code>pivot.rendererName</code> .....	4935
9.1.31	<code>pivot.showSubTotal</code> .....	4935
9.1.32	<code>pivot.add.buttonbar</code> .....	4936
9.1.33	<code>pivot.load.json.script.success</code> .....	4936
9.1.34	<code>pivot.load.json.script.error</code> .....	4936
9.1.35	<code>pivot.load.json.script.done</code> .....	4937
9.1.36	<code>pivot.load.json.script.complete</code> .....	4937
9.1.37	<code>pivot.json.path</code> .....	4938
9.1.38	<code>pivot.custom.script.after</code> .....	4938
<b>10</b>	<b>Spreadsheet [Options]</b> .....	<b>4939</b>
10.1	Spreadsheet Column .....	4939
10.1.1	<code>spreadsheet.column.object</code> .....	4939
10.1.2	<code>spreadsheet.column.n[].align</code> .....	4939
10.1.3	<code>spreadsheet.column.n[].dataType</code> .....	4940
10.1.4	<code>spreadsheet.customColumn.i[].format</code> .....	4940
10.1.5	<code>spreadsheet.column.n[].hidden</code> .....	4941
10.1.6	<code>spreadsheet.column.n[].render.object</code> .....	4941
10.1.7	<code>spreadsheet.column.n[].render.condition</code> .....	4942
10.1.8	<code>spreadsheet.column.n[].render.conditionTrue.style</code> .....	4942
10.1.9	<code>spreadsheet.column.n[].render.conditionFalse.style</code> .....	4943
10.1.10	<code>spreadsheet.column.n[].render.conditionTrue.style.fontStyle</code> .....	4943
10.1.11	<code>spreadsheet.column.n[].render.conditionFalse.style.fontStyle</code> .....	4944
10.1.12	<code>spreadsheet.column.n[].render.conditionTrue.style.color</code> .....	4944
10.1.13	<code>spreadsheet.column.n[].render.conditionFalse.style.color</code> .....	4944
10.1.14	<code>spreadsheet.column.n[].aggregate.type</code> .....	4945
10.1.15	<code>spreadsheet.column.n[].aggregate.edit</code> .....	4945
10.1.16	<code>spreadsheet.column.n[].title</code> .....	4946
10.1.17	<code>spreadsheet.customColumn.i[].title</code> .....	4946
10.1.18	<code>spreadsheet.customColumn.i[].value</code> .....	4947
10.1.19	<code>spreadsheet.customColumn.i[].aggregate.type</code> .....	4947
10.1.20	<code>spreadsheet.customColumn.i[].dataType</code> .....	4948
10.1.21	<code>spreadsheet.customColumn.i[].value.object</code> .....	4948
10.1.22	<code>spreadsheet.column.n[].render.condition.sameColumn</code> .....	4949

10.1.23	spreadsheet.column.n[].format .....	4949
10.1.24	spreadsheet.column.n[].clsHead .....	4949
10.1.25	spreadsheet.column.n[].cls .....	4950
10.1.26	spreadsheet.column.n[].tpCls .....	4950
10.1.27	spreadsheet.column.n[].halign .....	4951
	Erlaubte Werte .....	4951
10.1.28	spreadsheet.column.n[].hvalign .....	4951
	Erlaubte Werte .....	4952
10.1.29	spreadsheet.column.n[].valign .....	4952
	Erlaubte Werte .....	4952
10.1.30	spreadsheet.column.n[].width .....	4952
10.1.31	spreadsheet.column.n[].maxWidth .....	4953
10.1.32	spreadsheet.column.n[].minWidth .....	4953
10.1.33	spreadsheet.column.n[].menulcon .....	4954
10.1.34	spreadsheet.column.n[].menuUI .....	4954
10.1.35	spreadsheet.column.n[].menuInDisable .....	4955
10.1.36	spreadsheet.column.n[].menuInClose .....	4955
10.1.37	spreadsheet.column.n[].menuInHide .....	4956
10.1.38	spreadsheet.column.n[].nodrag .....	4956
10.1.39	spreadsheet.column.n[].nodrop .....	4956
10.1.40	spreadsheet.column.n[].resizable .....	4957
10.1.41	spreadsheet.column.n[].sortable .....	4957
10.1.42	spreadsheet.column.n[].type .....	4958
	Erlaubte Werte .....	4958
10.1.43	spreadsheet.column.n[].tpHide .....	4958
10.1.44	spreadsheet.column.n[].style .....	4959
10.1.45	spreadsheet.column.n[].styleHead .....	4959
10.1.46	spreadsheet.column.n[].aggregate.object .....	4960
10.2	<b>Spreadsheet Group Model .....</b>	<b>4960</b>
10.2.1	spreadsheet.groupModel.object .....	4960
10.2.2	spreadsheet.groupModel.dataIndx .....	4961
10.2.3	spreadsheet.groupModel.collapsed .....	4961
10.2.4	spreadsheet.groupModel.grandSummary .....	4962
10.2.5	spreadsheet.groupModel.agg .....	4962
10.2.6	spreadsheet.groupModel.fixCols .....	4963
10.2.7	spreadsheet.groupModel.groupCols .....	4963

10.2.8	spreadsheet.groupModel.header.....	4963
10.2.9	spreadsheet.groupModel.headerMenu .....	4964
10.2.10	spreadsheet.groupModel.icon .....	4964
10.2.11	spreadsheet.groupModel.indent .....	4965
10.2.12	spreadsheet.groupModel.menuItems .....	4965
10.2.13	spreadsheet.groupModel.merge.....	4966
10.2.14	spreadsheet.groupModel.nodeClose.....	4966
10.2.15	spreadsheet.groupModel.on .....	4967
10.2.16	spreadsheet.groupModel.pivot.....	4967
10.3	<b>Spreadsheet Spreadsheet Object .....</b>	<b>4967</b>
10.3.1	spreadsheet.rowInit.object .....	4967
10.3.2	spreadsheet.rowInit.condition.....	4968
10.3.3	spreadsheet.rowInit.conditionTrue.style.....	4968
10.3.4	spreadsheet.rowInit.conditionFalse.style.....	4969
10.3.5	spreadsheet.rowInit.conditionTrue.style.fontStyle.....	4969
10.3.6	spreadsheet.rowInit.conditionFalse.style.fontStyle.....	4970
10.3.7	spreadsheet.rowInit.conditionTrue.style.color .....	4970
10.3.8	spreadsheet.rowInit.conditionFalse.style.color .....	4971
10.3.9	spreadsheet.pageModel.enable.....	4971
10.3.10	spreadsheet.pivot.customColumn.i[].title .....	4972
10.3.11	spreadsheet.pivot.customColumn.i[].value.....	4972
10.3.12	spreadsheet.pivot.groupCols.dateColumn .....	4973
10.3.13	spreadsheet.pivot.groupCols.year.enable .....	4973
10.3.14	spreadsheet.pivot.groupCols.quarter.enable .....	4974
10.3.15	spreadsheet.pivot.groupCols.month.enable .....	4974
10.3.16	spreadsheet.aggregate.i[].name .....	4975
10.3.17	spreadsheet.aggregate.i[].definition .....	4976
10.3.18	spreadsheet.toolbar.enable .....	4976
10.3.19	spreadsheet.toolPanel.show.....	4977
10.3.20	spreadsheet.height .....	4977
10.3.21	spreadsheet.menuIcon .....	4977
10.3.22	spreadsheet.columnBorders.....	4978
10.3.23	spreadsheet.draggable .....	4978
10.3.24	spreadsheet.hwrap .....	4979
10.3.25	spreadsheet.maxHeight .....	4979
10.3.26	spreadsheet.maxWidth.....	4980



10.3.27	spreadsheet.minWidth .....	4980
10.3.28	spreadsheet.width .....	4980
10.3.29	spreadsheet.wrap .....	4981
10.3.30	spreadsheet.summaryOnTop.....	4981
10.3.31	spreadsheet.roundCorners .....	4982
10.3.32	spreadsheet.rowBorders .....	4982
10.3.33	spreadsheet.rowHt .....	4982
10.3.34	spreadsheet.rowHtHead .....	4983
10.3.35	spreadsheet.rowHtSum.....	4983
10.3.36	spreadsheet.showBottom .....	4984
10.3.37	spreadsheet.showHeader.....	4984
10.3.38	spreadsheet.showTitle .....	4984
10.3.39	spreadsheet.showTop .....	4985
10.3.40	spreadsheet.showToolbar.....	4985
10.3.41	spreadsheet.sortable.....	4986

**Hinweis:**

Die Beschreibung der Optionen gilt ab Dashlets-Plugin Version 2.2.5 .

Für ältere Versionen des Plugins können einzelne Optionen nicht verfügbar sein oder nur teilweise funktionieren.

## 1 C3 Charts [Options]

<b>Name</b>	C3 Charts
<b>Version</b>	0.4.11
<b>Hersteller</b>	C3JS
<b>Webseite</b>	<a href="https://c3js.org">https://c3js.org</a>

## 2 Dashlet Engine [Options]

<b>Name</b>	Dashlet Engine
<b>Version</b>	1.0
<b>Hersteller</b>	TRADUI Technologies GmbH
<b>Webseite</b>	<a href="https://tradui.de">https://tradui.de</a>

### 2.1 Dashlet Engine Buttons

#### 2.1.1 button.onclick

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Buttons
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die onclick-Funktion für jeden Button ist hier definiert.

Beispiel:

```
// Create the ButtonBar
buttonbar = dashlet.getDashlet().createDashletBar("DashletButtonBar", "TOP");
btn = buttonbar.addButton('Button 1', true);
btn.addOption("button.onclick", "console.log('Clicked Button')");
```

## 2.2 Dashlet Engine Data

### 2.2.1 data.script.dimZ1Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	filter, multiselect

Stellt die Werte für das zweite (Z1) Multiselect-Feld ein.

```
Beispiel:
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");

dashlet2.addOption("data.script.dimZ1Filter","multiselect2_getSelectOptions()");
```

### 2.2.2 data.script.template

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	scripts, template

Pfad zum entsprechenden Template Script.

Beispiel:

```
dashlet.addOption("data.script.template","js/echart/multiselect-dim-z2");
```

### 2.2.3 chart.predefined.xAxis

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multidimensional, vordefiniert

Definiert die Werte der X-Achse bei Charts mit Multiselect-Boxen.

Beispiel:

```
dashlet.addOption("chart.predefined.xAxis", "data1_dimX0");
```

## 2.2.4 data.script.searchOperator1

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.20
<b>Schlagworte / Tags</b>	scripts

Default : ==

Legt den Suchoperator für die Suche nach dem angegebenen Wert in der angegebenen Spalte fest. Akzeptierte Werte sind ==, equals, !=, notequals, <=, lesserthanorequalto, >=, greaterthanorqualto, <, lesserthan, >, greaterthan, contains, includes, notcontains, notincludes, startswith, notstartswith, endswith, notendswith

Beispiel:

```
dashlet.addOption("data.script.searchOperator1", "==");
```

### Erlaubte Werte

- equals
- notequals
- lesserthanorequalto
- lesserthanorequalto
- lesserthan
- lesserthan
- greaterthanorequalto

- greaterthan
- contains
- notcontains
- startswith
- notstartswith
- endswith
- notendswith

## 2.2.5 data.script.searchOperator2

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.20
<b>Schlagworte / Tags</b>	scripts

Default : ==

Legt den Suchoperator für die Suche nach dem angegebenen Wert in der angegebenen Spalte fest. Akzeptierte Werte sind ==, equals, !=, notequals, <=, lesserthanorequalto, >=, greaterthanorequalto, <, lesserthan, >, greaterthan, contains, includes, notcontains, notincludes, startswith, notstartswith, endswith, notendswith

Beispiel:

```
dashlet.addOption("data.script.searchOperator2", "==");
```

### Erlaubte Werte

- equals
- notequals
- lesserthan
- lesserthanorequalto
- greaterthan
- greaterthanorequalto
- contains
- notcontains
- startswith
- notstartswith
- endswith
- notendswith

## 2.2.6 data.script.searchOperator3

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.20
<b>Schlagworte / Tags</b>	scripts

Default : ==

Legt den Suchoperator für die Suche nach dem angegebenen Wert in der angegebenen Spalte fest. Akzeptierte Werte sind ==, equals, !=, notequals, <=, lesserthanorequalto, >=, greaterthanorequalto, <, lesserthan, >, greaterthan, contains, includes, notcontains, notincludes, startswith, notstartswith, endswith, notendswith

Beispiel:

```
dashlet.addOption("data.script.searchOperator3", "==");
```

### Erlaubte Werte

- equals
- notequals
- lesserthan
- lesserthanorequalto
- greaterthan
- greaterthanorequalto
- contains
- notcontains
- startswith
- notstartswith
- endswith
- notendswith

## 2.2.7 data.script.searchOperator4

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.20
<b>Schlagworte / Tags</b>	scripts

Default : ==



Legt den Suchoperator für die Suche nach dem angegebenen Wert in der angegebenen Spalte fest. Akzeptierte Werte sind ==, equals, !=, notequals, <=, lesserthanorequalto, >=, greaterthanorqualto, <, lesserthan, >, greaterthan, contains, includes, notcontains, notincludes, startswith, notstartswith, endswith, notendswith

Beispiel:

```
dashlet.addOption("data.script.searchOperator4", "==");
```

## Erlaubte Werte

- equals
- notequals
- lesserthan
- lesserthanorequalto
- greaterthan
- greaterthanorequalto
- contains
- notcontains
- startswith
- notstartswith
- endswith
- notendswith

## 2.2.8 data.script.searchOperator5

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.20
<b>Schlagworte / Tags</b>	scripts

Default : ==

Legt den Suchoperator für die Suche nach dem angegebenen Wert in der angegebenen Spalte fest. Akzeptierte Werte sind ==, equals, !=, notequals, <=, lesserthanorequalto, >=, greaterthanorqualto, <, lesserthan, >, greaterthan, contains, includes, notcontains, notincludes, startswith, notstartswith, endswith, notendswith

Beispiel:

```
dashlet.addOption("data.script.searchOperator5", "==");
```

## Erlaubte Werte

- equals

- notequals
- lesserthan
- lesserthanorequalto
- greaterthan
- greaterthanorequalto
- contains
- notcontains
- startswith
- endswith
- notstartswith
- notends

## 2.2.9 data.script.mode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	scripts

Gibt an, ob die Daten des MemoryBuffers in der HTML-Datei in einem Tabellen- oder Matrixformat gerendert werden sollen.

Beispiel:

```
dashlet.addOption("data.script.mode","table");
```

### Erlaubte Werte

- matrix
- table

## 2.2.10 data.script.dimZ0Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	filter, multiselect
---------------------------	---------------------

Stellt die Werte für das erste (Z0) Multiselect-Feld ein.

Example:

```
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");
```

### 2.2.11 data.script.dimZ2Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	filter, multiselect

Stellt die Werte für das dritte (Z2) Multiselect-Feld ein.

Beispiel:

```
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");
dashlet2.addOption("data.script.dimZ1Filter","multiselect2_getSelectOptions()");

dashlet2.addOption("data.script.dimZ2Filter","multiselect3_getSelectOptions()");
```

### 2.2.12 data.script.dimZ3Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	filter, multiselect

Stellt die Werte für das vierte (Z3) Multiselect-Feld ein.

Beispiel:

```
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");
dashlet2.addOption("data.script.dimZ1Filter","multiselect2_getSelectOptions()");
dashlet2.addOption("data.script.dimZ2Filter","multiselect3_getSelectOptions()");

dashlet2.addOption("data.script.dimZ3Filter","multiselect4_getSelectOptions()");
```

### 2.2.13 data.script.dimZ4Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	filter, multiselect

Stellt die Werte für das fünfte (Z4) Multiselect-Feld ein.

Beispiel:

```
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");
dashlet2.addOption("data.script.dimZ1Filter","multiselect2_getSelectOptions()");
dashlet2.addOption("data.script.dimZ2Filter","multiselect3_getSelectOptions()");
dashlet2.addOption("data.script.dimZ3Filter","multiselect4_getSelectOptions()");

dashlet2.addOption("data.script.dimZ4Filter","multiselect5_getSelectOptions()");
```

### 2.2.14 data.script.dimZ5Filter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	filter, multiselect

Stellen die Werte für das fünfte(Z4) Multiselect-Feld ein.

Beispiel:

```
dashlet2.addOption("data.script.dimZ0Filter","multiselect1_getSelectOptions()");
dashlet2.addOption("data.script.dimZ1Filter","multiselect2_getSelectOptions()");
dashlet2.addOption("data.script.dimZ2Filter","multiselect3_getSelectOptions()");
dashlet2.addOption("data.script.dimZ3Filter","multiselect4_getSelectOptions()");
dashlet2.addOption("data.script.dimZ4Filter","multiselect5_getSelectOptions()");
dashlet2.addOption("data.script.dimZ5Filter","multiselect6_getSelectOptions()");
```

## 2.2.15 chart.series.data.column

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	filter, multiselect

Definiert die Werte der Y-Achse bei Charts mit Multiselect-Boxen. Hier wird die 'Series'-Spalte definiert.

Beispiel:

```
dashlet.addOption("chart.series.data.column","value-dim-y");
```

## 2.2.16 chart.series.data.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	filter, multiselect

Funktion mit der die Series-Daten aus der Matrix gelesen werden. Diese ist in dem Template für das Data-Dashlet enthalten.

Beispiel:

```
dashlet.addOption("chart.series.data.function",
"dashlet.addOption('data2_getMatrixData('seriesidentifizierer')");
```

## 2.2.17 data.script.columnSearch

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.3
<b>Schlagworte / Tags</b>	column, datatable, filtercolumn

Legen Sie die Option fest, ein Suchfeld für die dataTable-Spalten anzuzeigen. Wird normalerweise mit den folgenden Optionen in Kombination verwendet.

```
Example:
// Column Search Filter
dashlet.addOption("data.script.columnSearch","true");
dashlet.addOption("data.script.columnSearchColumns","['CUSTOMERNUMBER',
'CUSTOMERNAME', 'ORDERNUMBER']");
dashlet.addOption("data.script.flexselect","true");
```

## 2.2.18 data.script.columnSearchColumns

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.3
<b>Schlagworte / Tags</b>	column, datatable, filtercolumn

Legen Sie die Option fest, ein Suchfeld für die selektierte dataTable-Spalten anzuzeigen. Wird normalerweise mit den folgenden Optionen in Kombination verwendet.

Example:

```
// Column Search Filter
dashlet.addOption("data.script.columnSearch","true");
dashlet.addOption("data.script.columnSearchColumns","['CUSTOMERNUMBER',
'CUSTOMERNAME', 'ORDERNUMBER']");
dashlet.addOption("data.script.flexselect","true");
```

## 2.2.19 data.script.flexselect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.3
<b>Schlagworte / Tags</b>	autocomplete, column, datatable, filtercolumn

Legen Sie die Option fest, ein Autocomplete - Suchfeld für die dataTable-Spalten anzuzeigen. Wird normalerweise mit den folgenden Optionen in Kombination verwendet.

Example:

```
// Column Search Filter
dashlet.addOption("data.script.columnSearch","true");
dashlet.addOption("data.script.columnSearchColumns","['CUSTOMERNUMBER',
'CUSTOMERNAME', 'ORDERNUMBER']");
dashlet.addOption("data.script.flexselect","true");
```

## 2.2.20 peity.chart.pie.fill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern der in einem Tabellen-Dashlet angezeigten Kreisdiagrammfarben (PieChart Farben)

Example:

```
dashlet.addOption("peity.chart.pie.fill",["#ff9900', '#fff4dd', '#ffc66e']");
```

### 2.2.21 peity.chart.pie.radius

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart , table

Option zum Ändern der in einem Tabellen-Dashlet angezeigten Kreisdiagramm Radius (PieChart Radius)

Example:

```
dashlet.addOption("peity.chart.pie.radius", "8");
```

### 2.2.22 peity.chart.line.delimiter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Trennzeichens für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.delimiter", "", "");
```

### 2.2.23 peity.chart.line.fill

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Farbe für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.fill", "#c6d9fd");
```

## 2.2.24 peity.chart.line.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	

Option zum Ändern des Höhes für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.height" ,"16");
```

## 2.2.25 peity.chart.line.min

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern der MinimumWerte für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.min" ,"0");
```

## 2.2.26 peity.chart.line.stroke

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Stroke-Farbe für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.stroke", "#4d89f9");
```

## 2.2.27 peity.chart.line.strokeWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Stroke-Widths für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.strokeWidth" ,"1");
```

## 2.2.28 peity.chart.line.width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern der Breite für die in einem Tabellen-Dashlet angezeigten Liniendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.line.width" ,"32");
```

## 2.2.29 peity.chart.bar.delimiter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Trennzeichens für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.delimiter","","");
```

## 2.2.30 peity.chart.bar.fill

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Farbe für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.fill", "#c6d9fd");
```

### 2.2.31 peity.chart.bar.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Höhes für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.height" ,"16");
```

### 2.2.32 peity.chart.bar.min

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern der MinimumWerte für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.min" ,"0");
```

### 2.2.33 peity.chart.bar.width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern der Breite für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.width" ,"32");
```

### 2.2.34 peity.chart.bar.padding

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.4
<b>Schlagworte / Tags</b>	peitychart

Option zum Ändern des Paddings für die in einem Tabellen-Dashlet angezeigten Balkendiagrammwerte

Example:

```
dashlet.addOption("peity.chart.bar.padding","0.1");
```

## 2.3 Dashlet Engine General

### 2.3.1 html.after.content

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen von HTML Markup innerhalb der Box nach dem Chart.

Praktisch um z.B. Inline-Dokumentation oder Fußnoten anzufügen.

### 2.3.2 html.before.content

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen von HTML Markup innerhalb der Box vor dem Chart.

Praktisch um z.B. Inline-Dokumentation oder Fußnoten anzufügen.

### 2.3.3 box.info.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	direkthilfe, help, hilfe, inline-documentation, inline-dokumentation, popup, steckbrief, symbol

Ermöglicht das Setzen eines Symbols für eine Direkthilfe.

Siehe auch: <https://confluence.tradui.net/x/JYBRaw>

und <https://fontawesome.com/icons>

### 2.3.4 box.info.tooltip.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert : hidden

Setzt die Sichtbarkeit eines Tooltips für die Direkthilfe.

Erlaubte Werte

- show
- hidden

### 2.3.5 box.info.tooltip.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Title des Tooltip Icons für die Direkthilfe.

### 2.3.6 box.info.tooltip.delay

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt das Animationsdelay des Tooltip Icons für die Direkthilfe (in millisekunden).

Format: JSON

Beispiel:

```
dashlet.addOption("box.info.tooltip.delay", '{"show": 500, "hide": 500}');
```

### 2.3.7 box.info.tooltip.template

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt des Tooltip Icons für die Direkthilfe.

### 2.3.8 box.info.popover.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert : hidden

Setzt die Sichtbarkeit eines Popovers für die Direkthilfe.

Erlaubte Werte

- show
- hidden

### 2.3.9 box.info.popover.placement

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die Position eines Popovers für die Direkthilfe.

Akzeptierter Wert: 'top', 'right', 'bottom', 'left'

### 2.3.10 box.info.popover.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Trigger für das Öffnen eines Popovers für die Direkthilfe.

Akzeptierte(r) Wert(e): click | hover | focus | manual

(mehrere Werte erlaubt, separieren durch Leerzeichen!)

### 2.3.11 box.info.popover.offset

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den relativen Offset eines Popovers für die Direkthilfe.

Akzeptierter Wert: 'x y'

Beispiel:

```
dashlet.addOption("box.info.popover.offset", "50 50");
```

### 2.3.12 box.info.popover.delay

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt das Animationsdelay des Popovers für die Direkthilfe (in millisekunden).

Format: JSON

Beispiel:

```
dashlet.addOption("box.info.popover.delay", "{\"show\": 500, \"hide\": 500}");
```

### 2.3.13 box.info.popover.html

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Inneres HTML im Popover zulassen.

### 2.3.14 box.info.popover.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Titel eines Popovers für die Direkthilfe.

### 2.3.15 box.info.popover.content

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt eines Popovers für die Direkthilfe. (HTML Markup erlaubt)

### 2.3.16 box.info.dialog.html

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Inneres HTML im Modal Dialog zulassen.

### 2.3.17 box.info.dialog.content

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt eines Modal Dialogs für die Direkthilfe. (HTML Markup erlaubt).

Beispiel:

```
dashlet.addOption("box.info.dialog.content", "<b>Hinweis:</b><br/><br/>Bitte auch unbedingt an diesen zweiten Hinweis halten!");
```

### 2.3.18 box.download.native.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Anzeigen eines Download Links zu einem generiertem Dokument.

Format ist: "<DisplayTitel>;<JobID>;<DateiName>"

Beispiel: "Mein Excel;123;mein\_excel.xlsx"

```
<span style="font-size:11.0pt">dashlet.addBoxOption("box.download.native.link",
"Download Native Excel;123;test.xlsx"); </span>
```

Nutze die Hilfsfunktion:

```
function generateDashletDownloadString(excelOutputManager) { ... }
```

Kann beliebig oft aufgerufen werden -> erzeugt mehrere Links als DropDown.

Siehe auch: <https://confluence.tradui.net/x/OYBRAw>

### 2.3.19 bar.n[].component.n[].preselect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagnworte / Tags</b>	multiselect, preselect, vorausfüllen

Ermöglicht eine Vorselektion von bestimmten Werten im Multiselect.

Sollte ein "nicht valider Wert" übergeben werden, wird dieser ignoriert.

Beispiel:

```
dashlet.addOption("bar.n[topbar1].component.n[multiselect2].preselect", "<strong><span
style="color:#2e75b6">[ 'Hardware', 'Software' ]</span></strong>");
```

### 2.3.20 chart.predefined.legend

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Ermöglicht die manuelle Festlegung der Anzeigereihenfolge von Legendenelementen. Die Reihenfolge der Legendenanzeige entspricht normalerweise der Reihenfolge im memoryBuffer. Diese Reihenfolge kann mit dieser Option geändert werden.

Beispiel:

```
dashlet.addOption("chart.predefined.legend", "data1_dimY0");
```

## 2.3.21 chart.custom.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit benutzerdefinierte Scripts / Funktionen einzufügen.

Beispiel:

```
dashlet.addOption("chart.custom.function", "function click_test(param)
{console.log(param);}");
dashlet.addOption("chart.event.click", "click_test");
```

## 2.3.22 childid

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	combination, combined, eine box, kombiniert, mehrere charts

Möglichkeit um Charts in einer Box gemeinsam zu rendern.

Beispiel:

```
dashlet.addOption("id","chart1");
dashlet.addOption("childid","chart2");
dashlet.addOption("childid","chart3");
```

### 2.3.23 javascript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen von benutzerdefinierten javascript Funktionen.

### 2.3.24 chart.event.click

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit ein Event zum Laden einer Tabelle beim Klicken auf die Chart-Komponente einzufügen.

```
dashlet.addOption("chart.event.click", "table1_load");
```

Die Funktion "table1\_load" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.25 table.lazy.load

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Auf true wird die Tabelle als Lazy-Load-Methode initialisiert. Dies wird normalerweise verwendet, um die Leistung des Programms zu verbessern.

### 2.3.26 table.lazy.load.time

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default: 1000ms

Legt die time-out Funktion für die Tabelle fest, die mit der Lazy-Load-Methode initialisiert werden soll. Wird normalerweise mit folgender Option verwendet:

```
dashlet.addOption("table.lazy.load", "true");
dashlet.addOption("table.lazy.load.time", "0");
```

### 2.3.27 chart.stacked

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht es, alle Komponenten wie in einem "Stacked-chart" zu stapeln.

```
dashlet.addOption("chart.stacked", "all");
```

### 2.3.28 chart.connect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Verbindung mehrerer Diagramme. Wird normalerweise in Kombination mit den folgenden Optionen verwendet:

```
// Connect von dashlets Hinzufügen
dashlet.addOption("chart.connect", "chart1->chart2;chart2->chart1");
dashlet.addOption("chart.connect.time.init", "100");
dashlet.addOption("chart.connect.time.retry", "3000");
```

Hinweis: Wenn die Diagramme mit der dataZoom-Komponente verbunden werden sollen, sollte die Komponente für alle zu verbindenden Diagramme definiert werden, und nicht nur für ein Diagramm.

### 2.3.29 chart.connect.time.init

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt die time-out-Funktion zum Verbinden mehrerer Diagramme fest. Wird normalerweise in Kombination mit den folgenden Optionen verwendet:

```
// Connect von dashlets Hinzufügen
dashlet.addOption("chart.connect", "chart1->chart2;chart2->chart1");
dashlet.addOption("chart.connect.time.init", "100");
dashlet.addOption("chart.connect.time.retry", "3000");
```

Hinweis: Wenn die Diagramme mit der dataZoom-Komponente verbunden werden sollen, sollte die Komponente für alle zu verbindenden Diagramme definiert werden, und nicht nur für ein Diagramm.

### 2.3.30 chart.connect.time.retry

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt die Zeitüberschreitungsfunktion für den erneuten Versuch beim Verbinden mehrerer Diagramme fest. Wird normalerweise in Kombination mit den folgenden Optionen verwendet:

```
// Connect von dashlets Hinzufügen
dashlet.addOption("chart.connect", "chart1->chart2;chart2->chart1");
dashlet.addOption("chart.connect.time.init", "100");
dashlet.addOption("chart.connect.time.retry", "3000");
```

Hinweis: Wenn die Diagramme mit der dataZoom-Komponente verbunden werden sollen, sollte die Komponente für alle zu verbindenden Diagramme definiert werden, und nicht nur für ein Diagramm.

### 2.3.31 chart.series.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	karte, map
---------------------------	------------

Definiert den Typ der Serie.

Akzeptierte Werte: "map", ...

### 2.3.32 chart.map.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Definiert den Typ der Karte. Der Name des Landes wird hier angegeben, damit die entsprechende JSON-Datei aus dem Ressourcenordner abgerufen werden kann.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
```

### 2.3.33 chart.map.contextMenu.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Legt fest, ob das Kontextmenü für Drilldown-Karten angezeigt werden soll oder nicht. Wird normalerweise in Kombination mit den folgenden Optionen verwendet:

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "germany");
dashlet.addOption("data.script.template", "js/maps/drillDownMaps/
germany_drilldownMap_withContextMenu");
dashlet.addOption("chart.map.contextMenu.show", "true");
dashlet.addOption("chart.event.contextmenu", "activate_drillDown");
```

### 2.3.34 chart.event.contextmenu

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wenn "activate\_drillDown" eingestellt ist, wird das Kontextmenü für DrillDown-Karten angezeigt. Wird normalerweise in Kombination mit den folgenden Optionen verwendet:

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "germany");
dashlet.addOption("data.script.template", "js/maps/drillDownMaps/
germany_drilldownMap_withContextMenu");
dashlet.addOption("chart.map.contextMenu.show", "true");
dashlet.addOption("chart.event.contextmenu", "activate_drillDown");
```

### 2.3.35 dashlet.external.json

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, die JSON-Dateien der Daten extern abzurufen

### 2.3.36 box.closetlink.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt die Sichtbarkeit der Schaltfläche "Close" für die Box fest.

Akzeptierter Wert: 'show', 'hidden'

Erlaubte Werte

- show
- hidden

### 2.3.37 box.collapselink.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt die Sichtbarkeit der Schaltfläche "Collapse/Expand" für die Box fest.

Akzeptierter Wert: 'show', 'hidden'

Erlaubte Werte

- show
- hidden

### 2.3.38 child.box

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	eine box

Wird normalerweise bei Gruppen-Dashlets verwendet, um die Box-Optionen für die untergeordneten Gruppen anzuzeigen.

Wert ist True: Dashlets mit Boxen werden mit Box und mit Bars gerendert

Wert ist False: Dashlets mit Boxen werden ohne Box aber mit Bars gerendert

Beispiel:

```
dashlet.addOption("child.box","true");
```

### 2.3.39 datatable.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	datatable, external-json-files

Für externe-JSON Aufrufe werden die einzelnen JSON-Dateien mit diesem Namen gespeichert. Der Aufruf zu den einzelnen JSON-Dateien wird auch mit diesem Namen aufgerufen.

Beispiel:

```
<span style="font-size:10.0pt"><span style="color:black">dashlet.setId(</span></span><span style="font-size:10.0pt"><span style="color:#2a00ff">"<strong>table 2</strong>"</span></span><span style="font-size:10.0pt"><span style="color:black">);</span></span><span style="font-size:10.0pt"><span style="color:#3f7f5f">//DataTable Id</span></span>
```

```
dashlet.addOption("dashlet.load.json.script.first", "table2_loadJson('data');\n"); //If no SeriesFilters are defined,
then the default group is 'data'
```

### 2.3.40 datatable.column.seriesfilter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	datatable

Die Spalte wird als Series-Filter definiert und das JSON Array wird mit dieser Spalte als 'Schlüssel' geformt.

Beispiel:

```
<span style="font-size:11.0pt">dashlet.addOption("datatable.column.seriesfilter", "Country");</span>
```

### 2.3.41 magic.add.buttonbar

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Fügt eine Buttonbar für den Seriesfilter hinzu

Beispiel:

```
// Buttons are displayed for each seriesFilter value
```

```
dashlet.addOption("datatable.column.seriesfilter", "Productline");
dashlet.addOption("magic.add.buttonbar", "seriesfilter");
dashlet.addOption("magic.buttonbar.event.onclick",
"table1_clear();table1_loadJson('%SERIESNAME%');");
```

### 2.3.42 magic.buttonbar.event.onclick

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die onclick-Funktion für jeden Button ist hier definiert.

Beispiel:

```
dashlet.addOption("magic.add.buttonbar", "seriesfilter");
dashlet.addOption("magic.buttonbar.event.onclick",
"table1_clear();table1_loadJson('%SERIESNAME%');");
```

### 2.3.43 chart.inline.mode

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	

Default Wert: false

Aktiviert die Möglichkeit, ein EChart ohne Boxen zu rendern. Zum Beispiel um ein EChart in einer KPI-Box anzuzeigen.

Beispiel:

```
<span style="font-size:11.0pt">dashlet.addOption("chart.inline.mode", "true");</span>
```

### 2.3.44 bar.n[].component.n[].options.var

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect, vorausfüllen

Legt die Werte für die MultiSelect-Box fest.

Beispiel:

```
dashlet.addOption("bar.n[topbar1].component.n[multiselect2].options.var",
"daa1_dimZ1");
```

### 2.3.45 bar.n[].component.onSelectAll

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect, onselectall

Funktionsdefinition, wenn auf den Wert 'Alle auswählen' im MultiSelect-Feld geklickt wird.

Beispiel:

```
dashlet.addOption("bar.n[topbar1].component.onSelectAll",
"chart1.clear();chart1_echartOption.series =
chart1_getPredefinedSeriesData().data1;chart1.setOption(chart1_echartOption);");
```

### 2.3.46 bar.n[].component.onChange

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect, onchange

Funktionsdefinition, wenn ein Wert aus den MultiSelect-Box-Optionen ausgewählt wird.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.onChange",
"chart1.clear();chart1_echartOption.series =
chart1_getPredefinedSeriesData().data1;chart1.setOption(chart1_echartOption);");
```

### 2.3.47 bar.n[].component.onDeselectAll

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect, ondeselectall

Funktionsdefinition, wenn alle Werte der MultiSelect-Box abgewählt werden.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.onDeselectAll", "chart1.clear();chart1_echartOption.series =
chart1_getPredefinedSeriesData().data1;chart1.setOption(chart1_echartOption);");
```

### 2.3.48 bar.n[].block

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar

<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	barstyle

Legt den Blockstil der oberen Leisten fest. Standardmäßig werden die Leisten nebeneinander gerendert.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].block", "span");
```

### Erlaubte Werte

- div
- span

## 2.3.49 bar.n[].component.n[].maxHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die maximale Höhe des Dropdowns. Dies ist nützlich, wenn die Liste viele Optionen enthält.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].maxHeight", "200");
```

## 2.3.50 bar.n[].component.n[].enableFiltering

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar

<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Auf true oder false setzen, um den Filter zu aktivieren oder zu deaktivieren. Ein Filtereingang wird hinzugefügt, um alle Optionen dynamisch zu filtern.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].enableFiltering","true");
```

### 2.3.51 bar.n[].component.n[].enableCaseInsensitiveFiltering

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Auf true oder false setzen, um den Filter zu aktivieren oder zu deaktivieren. Ein Filtereingang wird hinzugefügt, um alle Optionen dynamisch zu filtern. Bei der Suche wird die Groß- und Kleinschreibung nicht berücksichtigt.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].enableCaseInsensitiveFiltering","true");
```

### 2.3.52 bar.n[].component.n[].enableFullValueFiltering

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	TESTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Auf true setzen, um die vollständige Wertfilterung zu aktivieren. Das heißt, alle Optionen werden angezeigt, bei denen die Abfrage ein Präfix von ihnen ist.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].enableFiltering", "true");
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].enableFullValueFiltering", "true");
```

### 2.3.53 bar.n[].component.n[].filterBehavior

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die Optionen werden basierend auf ihrem Text gefiltert. Dieses Verhalten kann geändert werden, um den Wert der Optionen oder sowohl den Text als auch den Wert zu verwenden.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].filterBehavior", "'value'");
```

### 2.3.54 bar.n[].component.n[].filterPlaceholder

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	multiselect
---------------------------	-------------

Der für die Filtereingabe verwendete Platzhalter.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].filterPlaceholder",
  "'search'");
```

### 2.3.55 bar.n[].component.n[].buttonText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Callback (Rückruf), der den auf der Schaltfläche angezeigten Text angibt, abhängig von den aktuell ausgewählten Optionen.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].buttonText", "function(options, select) {n" +
"      if (options.length === 0) {n" +
"        return 'No option selected ...';n" +
"      }n" +
"      else if (options.length > 3) {n" +
"        return 'More than 3 options selected!';n" +
"      }n" +
"      else {n" +
"        var labels = [];n" +
"        options.each(function() {n" +
"          if ($(this).attr('label') !== undefined) {n" +
"            labels.push($(this).attr('label'));n" +
"          }n" +
"          else {n" +
"            labels.push($(this).html());n" +
"          }n" +
"        });n" +
"        return labels.join(', ') + ';n" +
"      }n" +
"    });
```

### 2.3.56 bar.n[].component.n[].buttonTitle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Rückruf (callback), der den Titel der Schaltfläche angibt.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].buttonTitle", "function(options, select) {n" +
"    var labels = [];n" +
"    options.each(function () {n" +
"        labels.push($(this).text());n" +
"    });n" +
"    return labels.join(' - ');n" +
"}");
```

### 2.3.57 bar.n[].component.n[].nonSelectedText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der Text, der angezeigt wird, wenn keine Option ausgewählt ist.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].nonSelectedText", "none selected");
```

### 2.3.58 bar.n[].component.n[].nSelectedText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der Text, der angezeigt wird, wenn mehr als numberDisplayed-Optionen ausgewählt sind.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].nSelectedText", "many selected");
```

### 2.3.59 bar.n[].component.n[].allSelectedText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

allSelectedText ist der Text, der angezeigt wird, wenn alle Optionen ausgewählt sind.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].allSelectedText", "All selected");
```

### 2.3.60 bar.n[].component.n[].optionClass

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General



<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Callback (Rückruf) zum Definieren der Klassen für die (ListItem) li-Elemente, die Check-Boxen und Beschriftungen enthalten.

### 2.3.61 bar.n[].component.n[].selectedClass

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die Klasse(n), die auf ausgewählte Optionen angewendet werden.

### 2.3.62 bar.n[].component.n[].includeSelectAllOption

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Auf true oder false setzen, um die "Alle auswählen"-Option zu aktivieren oder zu deaktivieren.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].includeSelectAllOption", "true");
```

### 2.3.63 bar.n[].component.n[].enableClickableOptGroups

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Wenn true festgelegt ist, können optgroups angeklickt werden, sodass auf einfache Weise mehrere Optionen ausgewählt werden können, die zu derselben Gruppe gehören.

### 2.3.64 bar.n[].component.n[].enableCollapsibleOptGroups

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Wenn true festgelegt ist, können Optgruppen zusammengefasst werden.

### 2.3.65 bar.n[].component.n[].disableIfEmpty

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Wenn true festgelegt ist, wird die Mehrfachauswahl deaktiviert, wenn keine Optionen angegeben sind.

### 2.3.66 bar.n[].component.n[].disabledText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der angezeigte Text, wenn die Mehrfachauswahl deaktiviert ist. Beachten Sie, dass diese Option standardmäßig auf die leere Zeichenfolge " gesetzt ist. Das heißt, nonSelectedText wird angezeigt, wenn die Mehrfachauswahl deaktiviert ist und keine Optionen ausgewählt sind.

### 2.3.67 bar.n[].component.n[].dropRight

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Das Dropdown-Menü wird auf der rechten Seite angezeigt.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].dropRight","true");
```

### 2.3.68 bar.n[].component.n[].dropUp

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Das Dropdown-Menü wird oben angezeigt.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].dropUp","true");
```

### 2.3.69 bar.n[].component.n[].checkboxName

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der Name, der für die generierten CheckBoxes verwendet wird.

### 2.3.70 bar.n[].component.n[].onInitialized

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Eine Funktion, die ausgelöst wird, wenn die Mehrfachauswahl initialisiert ist.

### 2.3.71 bar.n[].component.n[].onDropdownShow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Rückruf (callback), der aufgerufen wird, wenn das Dropdown angezeigt wird.

### 2.3.72 bar.n[].component.n[].onDropdownHide

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Rückruf (callback), der aufgerufen wird, wenn das Dropdown geschlossen wird.

### 2.3.73 bar.n[].component.n[].onDropdownShown

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Rückruf (callback), der aufgerufen wird, nachdem das Dropdown angezeigt wurde.

### 2.3.74 bar.n[].component.n[].onDropdownHidden

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Ein Rückruf (callback), der aufgerufen wird, nachdem das Dropdown geschlossen wurde.

### 2.3.75 bar.n[].component.n[].buttonClass

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die Klasse der Multiselect-Schaltfläche.

### 2.3.76 bar.n[].component.n[].inheritClass

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Erbt die Klasse der Schaltfläche von der ursprünglichen Auswahl.

### 2.3.77 bar.n[].component.n[].buttonContainer

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der Container der sowohl die Schaltfläche als auch das Dropdown enthält.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].buttonContainer","<div class='btn-group' />");
```

### 2.3.78 bar.n[].component.n[].buttonWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die Breite der Mehrfachauswahlschaltfläche kann mit dieser Option festgelegt werden. Tatsächlich beschreibt buttonWidth die Breite des Containers '.btn-group' und die Breite der Schaltfläche ist auf 100% festgelegt. .

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].buttonWidth","200px");
```

### 2.3.79 bar.n[].component.n[].numberDisplayed

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Diese Option wird von den Funktionen `buttonText` und `buttonTitle` verwendet, um zu bestimmen, ob zu viele Optionen angezeigt werden würden. Die Funktionalität kann deaktiviert werden, indem `numberDisplayed` auf 0 gesetzt wird.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].numberDisplayed", "1");
```

### 2.3.80 `bar.n[].component.n[].delimiterText`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Legt das Trennzeichen für die Liste der ausgewählten Elemente für das Mouseover fest. Der Standardwert ist `'`. Stellen Sie `\n` für eine übersichtlichere Anzeige ein.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].delimiterText", "' ; '");
```

### 2.3.81 `bar.n[].component.n[].optionLabel`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	multiselect
---------------------------	-------------

Ein callback (Rückruf) zum Definieren der Beschriftungen der Optionen.

### 2.3.82 bar.n[].component.n[].selectAllJustVisible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Wenn Sie `includeSelectAllOption` und `enableFiltering` auf `true` setzen, wählt die Option Alle auswählen immer nur die sichtbaren Optionen aus. Wenn Sie `selectAllJustVisible` auf `false` setzen, wird dieses Verhalten so geändert, dass immer alle Optionen (unabhängig davon, ob sie sichtbar sind) ausgewählt werden.

### 2.3.83 bar.n[].component.n[].selectAllText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Der Text, der für die Option "Alle auswählen" angezeigt wird.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].component.n[multiselect1].selectAllText", "'SELECT ALL '");
```

### 2.3.84 bar.n[].component.n[].selectAllValue

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Die Select-All Option wird als zusätzliche Option innerhalb der Auswahl hinzugefügt. Um diese Option von den ursprünglichen Optionen zu unterscheiden, kann der für die Option "Alle auswählen" verwendete Wert mit der Option selectAllValue konfiguriert werden.

### 2.3.85 bar.n[].component.n[].selectAllName

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Mit dieser Option können Sie den Namen steuern, der der Option "Alle auswählen" zugewiesen wird

### 2.3.86 bar.n[].component.n[].selectAllNumber

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	multiselect

Bei der Einstellung true (Standard) wird die Anzahl der ausgewählten Optionen in Klammern angezeigt, wenn alle Optionen ausgewählt sind.

### 2.3.87 bar.n[].style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	barstyle

Legt den Stil der oberen Leisten fest.

Beispiel:

```
dashlet3.addOption("bar.n[topbar1].style", "margin-left: 20px;");
```

### 2.3.88 chart.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt die Höhe des Chart fest.

Beispiel:

```
dashlet.addOption("chart.height", "700px");
```

### 2.3.89 table.customScript

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	datatable

Option zum Hinzufügen eines benutzerdefinierten Skripts für die DataTable-Komponente

Beispiel:

```
dashlet.addOption("table.customScript",
"$ (document).ready(function(){n" +
"   var table = $('#table1').DataTable();n" +
"   table.$('tr').css("cursor", "pointer");n" +
"   $('#table1 tbody').on( 'click', 'tr', function () {n" +
"       alert('Click on table row: ' + table.rows(this).indexes()[0] + '\\ndata: [' +
table.rows(this).data()[0] + '']);n" +
"   } );n" +
"});");
```

### 2.3.90 box.info.dialog.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Info-Dialogfelds fest

Erlaubte Werte

- show
- hidden

### 2.3.91 box.info.dialog.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Setzt den Titel eines Modal Dialogs für die Direkthilfe.

Beispiel:

```
dashlet.addOption("box.info.dialog.title", "Kleiner <span style="color:#0000FF;">Dialog-Hinweis</span>");
```

### 2.3.92 box.info.dialog.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt den Titel für das Info-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.info.dialog.type", "small");
```

Erlaubte Werte

- small
- default
- large

### 2.3.93 box.info.dialog.footer.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Info-Dialogfeld fest

Erlaubte Werte

- show
- hidden

### 2.3.94 box.info.dialog.footer.button.close.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Info-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.info.dialog.footer.button.close.title", "Hinweis schließen");
```

### 2.3.95 box.alert.dialog.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8

**Schlagworte / Tags**

Legt die Option zum Anzeigen des Alert-Dialogfelds fest

## Erlaubte Werte

- show
- hidden

### 2.3.96 box.alert.dialog.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Setzt den Titel eines Modal Dialogs für die Alert-Box

Beispiel:

```
dashlet.addOption("box.alert.dialog.title", "Alert");
```

### 2.3.97 box.alert.dialog.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt den Titel für das Alert-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.alert.dialog.type", "small");
```

## Erlaubte Werte

- small
- default
- large

### 2.3.98 box.alert.dialog.content

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Setzt den Inhalt eines Modal Dialogs für die Alert-Box. (HTML Markup erlaubt).

Beispiel:

```
dashlet.addOption("box.alert.dialog.content", "empty");
```

### 2.3.99 box.alert.dialog.footer.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Alert-Dialogfeld fest

## Erlaubte Werte

- show
- hidden



### 2.3.100 box.alert.dialog.footer.button.close.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Alert-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.alert.dialog.footer.button.close.title", "Schließen");
```

### 2.3.101 box.confirm.dialog.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Confirm-Dialogfelds fest

Erlaubte Werte

- show
- hidden

### 2.3.102 box.confirm.dialog.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box

<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Setzt den Titel eines Modal Dialogs für die Confirm-Box

Beispiel:

```
dashlet.addOption("box.confirm.dialog.title", "Confirm");
```

### 2.3.103 box.confirm.dialog.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt den Titel für das Confirm-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.confirm.dialog.type", "small");
```

#### Erlaubte Werte

- small
- default
- large

### 2.3.104 box.confirm.dialog.content

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Setzt den Inhalt eines confirm Modal Dialogs für die Direkthilfe.

Beispiel:

```
dashlet.addOption("box.confirm.dialog.content", "empty");
```

### 2.3.105 box.confirm.dialog.footer.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Confirm-Dialogfeld fest

Erlaubte Werte

- show
- hidden

### 2.3.106 box.confirm.dialog.footer.button.close.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen des Footers für die Confirm-Dialogfeld fest.

Beispiel:

```
dashlet.addOption("box.confirm.dialog.footer.button.close.title", "Schließen");
```

### 2.3.107 box.confirm.dialog.footer.button.cancel.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt die Option zum Anzeigen ein Cancel Button an das Footer für die Confirm-Dialogfeld fest

Erlaubte Werte

- show
- hidden

### 2.3.108 box.confirm.dialog.footer.button.cancel.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.9
<b>Schlagworte / Tags</b>	

Legt die Option fest, den Titel der Cancel-Button für ein Confirm-Dialogfeldanzuzeigen.

Beispiel:

```
dashlet.addOption("box.confirm.dialog.footer.button.cancel.title", "Cancel");
```

### 2.3.109 dashlet.load.json.script.first

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, die initial JSON-Dateien von ein extern anzuzeigen

Example:

```
//external JSON
dashlet.setId("table2");//DataTable Id
dashlet.addOption("datatable.id", "classicModels");// Id für die Datatable und Files für externe Aufruf. Die JSON-Datei ist mit diese Name gespeichert. Falls nicht gesetzt, dann nimmt die Dashlet-Plugin ein Default Wert.
dashlet.addOption("dashlet.external.json", "true");//Render the data as an external JSON file
dashlet.addOption("dashlet.load.json.script.first", "table2_loadJson('data');\n");//If no SeriesFilters are defined, then the default group is 'data'
```

### 2.3.110 dashlet.json.path

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, den Pfad zu den externen JSON-Dateien anzugeben.

Beispiel:

```
// set path for working in servBIRD
// set the path for the external JSON files
glJobID = 0;
var appContext = reportContext.getAppContext(); // HashMap
if (appContext != null && appContext.containsKey("REPORT_JOBID"))
{
    glJobID = appContext.get("REPORT_JOBID");
}
dashlet.addOption("dashlet.json.path", "http://localhost:8080/servBIRD/
streamOutputFile?jobid=" + glJobID + "&filename=d/");
```

### 2.3.111 chart.custom.init.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Möglichkeit die benutzerdefinierte Initial Aufruf (z.b. für ein Gantt-chart) Scripts / Funktionen einzufügen.

Beispiel:

```
dashlet.addOption("chart.custom.init.function" , "chart1_LazyLoad(chart1_dataSet,
chart1_dataSet_dim);");
```

### 2.3.112 bar.n[].component.n[].initbuttons

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	multiselect

Die Liste der Schaltflächen, die beim Ausführen des Dashboards angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("bar.n[bottombar1].component.n[piselect1].initbuttons",["'Button 1', 'Button 2'"]);
```

### 2.3.113 bar.n[].component.n[].onRemoveButton

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	multiselect

Die Funktionsdefinition, die beim Löschen einer Schaltfläche aufgerufen werden soll.

Beispiel:

```
dashlet.addOption("bar.n[bottombar1].component.n[piselect1].onRemoveButton", "function(params) {chart1_piselect_remove_callback(params)}");
```

### 2.3.114 bar.n[].component.n[].onAddButton

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	multiselect

Die Funktionsdefinition, die beim Erstellen einer Schaltfläche aufgerufen werden soll.

Beispiel:

```
dashlet.addOption("bar.n[bottombar1].component.n[piselect1].onAddButton", "function(params) {chart1_piselect_add_callback(params)}");
```

### 2.3.115 html

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen von HTML-Skripten.

### 2.3.116 box.header.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Styl eines Headers

Beispiel:

```
dashlet.addOption("box.header.style", "border-bottom-color:
rgba(127,127,127,0.5);border-width: 0px 0px 2px;");
```



### 2.3.117 box.title.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Titel-Styl eines Boxes.

Beispiel:

```
dashlet.addOption("box.title.style", "font-size: 0.9em;color: red;");
```

### 2.3.118 box.subtitle.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Subtitel-Styl eines Boxes.

Beispiel:

```
dashlet.addOption("box.subtitle.style", "font-size: 0.9em;color: red;");
```

### 2.3.119 box.content.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	content, styling

Möglichkeit zum stylen des Inhalts

Beispiel:

```
dashlet.addOption("box.content.style","height: 120px;");
```

### 2.3.120 box.block.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling

Möglichkeit zum stylen des Blocks

Beispiel:

```
dashlet.addOption("box.block.style","padding: 0px;");
```

### 2.3.121 box.headerbar.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling

Möglichkeit zum stylen des BlocHeaderBarks

Beispiel:

```
dashlet.addOption("box.headerbar.style" , "overflow: hidden; border-bottom-color: rgba(127,127,127,0.5);border: none; margin-top: 12px; margin-bottom: 12px;");
```

### 2.3.122 box.headerbar.html

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Inneres HTML im Box Header zulassen.

Beispiel:

```
dashlet.addOption("box.headerbar.html" , "<b>TEST</b>");
```

### 2.3.123 chart.predefined.series

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzen der Chart Series

Beispiel:

```
dashlet.addOption("chart.predefined.series" , "Norway,Germany,USA,Italy,Quatsch");
```

### 2.3.124 chart.predefined.legend.use.series

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht die manuelle Festlegung der Anzeigereihenfolge von Legendenelementen. Die Reihenfolge der Legendenanzeige entspricht normalerweise der Reihenfolge im memoryBuffer. Diese Reihenfolge kann mit dieser Option geändert werden.

Beispiel:

```
dashlet.addOption("chart.predefined.legend.use.series", "true");
```

### 2.3.125 chart.custom.openpanel.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.3
<b>Schlagworte / Tags</b>	gantt

Möglichkeit die benutzerdefinierte Initial Aufruf (z.b. für ein Gantt-chart) Scripts / Funktionen einzufügen für das Chart in ein ModalBox (Popup Fenster) zu öffnen.

Beispiel:

```
dashlet.addOption("chart.custom.openpanel.function" , "var _temp_dataSet =
chart1_dataSet;n" +
    "if (typeof chart1_filtered_DataSet !== 'undefined' &&
chart1_filtered_DataSet != null) _temp_dataSet = chart1_filtered_DataSet;n" +
    "chart1_LazyLoad(_temp_dataSet, chart1_dataSet_dim);");

// Entfernen die chart.custom.init.function Option
// dashlet.addOption("chart.custom.init.function" , "chart1_LazyLoad(chart1_dataSet,
chart1_dataSet_dim);");
```

### 2.3.126 responsive.grid

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	
<b>Schlagworte / Tags</b>	

Legt die Option für die Reaktionsfähigkeit der Dashlet fest. Die erste Wert ist for die 'Large' Display units, die zweite ist for 'Medium' Display units, die dritte für 'Small' display units und die vierte für 'Extra-Small' Display units.

Beispiel:

```
dashlet.addOption("responsive.grid", "12-12-12");
```

### 2.3.127 title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	title

Frei als Titel definierbarer Text.

Beispiel:

```
dashlet.addOption("title", "Products");
```

### 2.3.128 subtitle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	subtitle

Frei als Subtitel definierbarer Text.

Beispiel:

```
dashlet.addOption("subtitle", "Details");
```

### 2.3.129 chart.event.dataviewchanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ändern des Ereignisses (Event) des Datenansichtstools (Data view tool) in der Toolbox.

```
dashlet.addOption("chart.event. dataviewchanged ", "dataViewFunction");
```

Die Funktion "*dataViewFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function dataViewFunction(param)
{console.log(params);}");
```

### 2.3.130 chart.event.axisareaselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bei der Auswahl des Achsenbereichs kann die folgende Methode verwendet werden, um Datenindizes von aktuell hervorgehobenen Linien abzurufen. Dies ist die Liste der Indizes in Daten von Serien.

```
dashlet.addOption("chart.event.axisareaselected", "AxisFunction");
```

Die "AxisFunction" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function AxisFunction(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.131 chart.event.brush

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ereignis (Event) ausgelöst, nachdem die Aktionsbürste ausgelöst wurde.

```
dashlet.addOption("chart.event.brush", "brushFunction");
```

Die Funktion "brushFunction" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function brushFunction(params)
{console.log(params);}");
```

### 2.3.132 chart.event.brushselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beachten was ausgewählt ist. Dieses Ereignis wird ausgelöst, wenn dispatchAction aufgerufen wird oder das Pinselverhalten verwendet wird. Dieses Ereignis wird jedoch in setOption nicht ausgelöst.

```
dashlet.addOption("chart.event.brushselected", "brushFunction");
```

Die Funktion "*brushFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function brushFunction(params)
{console.log(params);}");
```

### 2.3.133 chart.event.datazoom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event), das nach dem Zoomen des Datenbereichs ausgegeben wird.

```
dashlet.addOption("chart.event.datazoom", "dataZoomFunction");
```



Die Funktion "*dataZoomFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function dataZoomFunction(param)
{console.log(params);}");
```

### 2.3.134 chart.event.datarangeselectd

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event), das ausgegeben wird, nachdem der Bereich in visualMap geändert wurde.

```
dashlet.addOption("chart.event.datarangeselectd ", "dataRangeFunction");
```

Die Funktion "*dataRangeFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function dataRangeFunction(param)
{console.log(params);}");
```

### 2.3.135 chart.event.dbclick

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Möglichkeit ein Event zum Laden einer Tabelle beim doppelklicken auf die Chart-Komponente einzufügen.

```
dashlet.addOption("chart.event.dblclick", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.136 chart.event.finished

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Wird ausgelöst, wenn das Rendern beendet ist, dh wenn die Animation beendet und das progressive Rendern beendet ist.

```
dashlet.addOption("chart.event.finished", "finishedFunction");
```

Die Funktion "*finishedFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function finishedFunction(param)
{console.log(params);}");
```

### 2.3.137 chart.event.focusnodeadjacency

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	event
---------------------------	-------

Markieren (Highlight) den angegebenen Knoten und alle angrenzenden Knoten.

```
dashlet.addOption("chart.event.focusnodeadjacency ", "focusNodeFunction");
```

Die Funktion "*focusNodeFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function focusNodeFunction(param)
{console.log(params);}");
```

### 2.3.138 chart.event.globalcursortaken

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Der Schalter der Bürste. Durch diese Aktion kann die Maus zum Bürsten aktiviert / deaktiviert werden. Tatsächlich verwenden die Pinselschaltflächen in der Toolbox nur diese Funktion.

```
dashlet.addOption("chart.event.globalcursortaken", "globalCursorFunction");
```

Die Funktion "*globalCursorFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function globalCursorFunction(param)
{console.log(params);}");
```

### 2.3.139 chart.event.mousedown

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen eines Ereignisses für das Mousedown-Ereignis

```
dashlet.addOption("chart.event.mousedown", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.140 chart.event.mousemove

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen eines Ereignisses für das mousemove -Ereignis

```
dashlet.addOption("chart.event.mousemove", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.141 chart.event.mouseout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen eines Ereignisses für das mouseout -Ereignis

```
dashlet.addOption("chart.event.mouseout", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.142 chart.event.mouseover

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ermöglicht das Einfügen eines Ereignisses für das mouseover -Ereignis

```
dashlet.addOption("chart.event.mouseover", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.143 chart.event.mouseup

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ermöglicht das Einfügen eines Ereignisses für das mouseup-Ereignis

```
dashlet.addOption("chart.event.mouseup", "table1_load");
```

Die Funktion "table1\_load" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.144 chart.event.globalout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Einfügen eines Ereignisses für das globalout -Ereignis

```
dashlet.addOption("chart.event.globalout", "table1_load");
```

Die Funktion "*table1\_load*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function table1_load(param)
{table1_lazyLoadTable(param.name,3);}");
```

### 2.3.145 chart.event.legendselectchanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ereignis (Event), das ausgegeben wird, nachdem die Legende Statusänderungen ausgewählt hat.

Achtung: Dieses Ereignis wird ausgegeben, wenn Benutzer die Legendenschaltfläche in der Legendenkomponente umschalten.

```
dashlet.addOption("chart.event.legendselectchanged", "legendChange");
```

Die Funktion "*legendChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function legendChange(param)
{console.log(param);}");
```

### 2.3.146 chart.event.legendselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event), das ausgegeben wird, nachdem die Legende ausgewählt wurde. .

```
dashlet.addOption("chart.event.legendselected", "legendChange");
```

Die Funktion "*legendChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function legendChange(param)
{console.log(param);}");
```

### 2.3.147 chart.event.legendunselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event), das nach Aufheben der Auswahl der Legende ausgegeben wird.

```
dashlet.addOption("chart.event.legendunselected", "legendChange");
```

Die Funktion "*legendChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function legendChange(param)
{console.log(param);}");
```

### 2.3.148 chart.event.legendselectall

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ereignis (Event), das ausgegeben wird, nachdem alle Legenden ausgewählt wurden.

```
dashlet.addOption("chart.event.legendselectall", "legendChange");
```

Die Funktion "*legendChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function legendChange(param)
{console.log(param);}");
```

### 2.3.149 chart.event.legendscroll

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event) beim Scrollen der Triggerlegende.

```
dashlet.addOption("chart.event.legendscroll", "legendChange");
```

Die Funktion "*legendChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function legendChange(param)
{console.log(param);}");
```

### 2.3.150 chart.event.unfocusnodeadjacency

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Benachbarte Knoten heben das Ereignis im Diagramm rückwärts hervor.

```
dashlet.addOption("chart.event.unfocusnodeadjacency", "focusNodeFunction");
```

Die Funktion "*focusNodeFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function focusNodeFunction(param)
{console.log(params);}");
```

### 2.3.151 chart.event.magictypechanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Wechselergebnis (Event) des Magic - Werkzeugs in der Toolbox.

```
dashlet.addOption("chart.event.magictypechanged", "magicChange");
```

Die Funktion "*magicChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function magicChange(param)
{console.log(param);}");
```

### 2.3.152 chart.event.mapselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

map regionEvent nach Auswahl.

Die Verwendung von dispatchAction kann dieses Ereignis auslösen, aber ein Benutzer, der auf dieses Ereignis klickt, wird dies nicht auslösen (Benutzer, der auf ein Ereignis klickt, verwendet bitte mapselectchanged).

```
dashlet.addOption("chart.event.mapselected", "mapChange");
```

Die Funktion "*mapChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function mapChange(param)
{console.log(param);}");
```

### 2.3.153 chart.event.mapunselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Kartenregion bricht das ausgewählte Ereignis (Event) ab.

Mit dispatchAction wird dieses Ereignis ausgelöst, durch Klicken des Benutzers wird es jedoch nicht ausgelöst. (Informationen zum Klicken auf ein Ereignis finden Sie unter mapselectchanged).

```
dashlet.addOption("chart.event.mapunselected", "mapChange");
```

Die Funktion "*mapChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function mapChange(param)
{console.log(param);}");
```

### 2.3.154 chart.event.mapselectchanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Ereignis (Event), das ausgegeben wird, nachdem der Kartenbereich Statusänderungen ausgewählt hat. Es wird ausgelöst, wenn der Benutzer zur Auswahl klickt.

```
dashlet.addOption("chart.event.mapselectchanged", "mapChange");
```

Die Funktion "*mapChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function mapChange(param)
{console.log(param);}");
```

### 2.3.155 chart.event.pieselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Kreisdiagramm Event nach Auswahl.

Die Verwendung von `dispatchAction` kann dieses Ereignis auslösen, aber ein Benutzer, der auf dieses Ereignis klickt, wird dies nicht auslösen (Benutzer, der auf ein Ereignis klickt, verwendet bitte `pieselectchanged`).

```
dashlet.addOption("chart.event.pieselected", "pieChange");
```

Die Funktion "pieChange" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function pieChange(param)
{console.log(param);}");
```

### 2.3.156 chart.event.pieunselected

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Kreisdiagramm bricht das ausgewählte Ereignis ab.

Mit dispatchAction wird dieses Ereignis ausgelöst, durch Klicken des Benutzers wird es jedoch nicht ausgelöst. (Informationen zum Klicken auf ein Ereignis finden Sie unter pieselectchanged).

```
dashlet.addOption("chart.event.pieunselected", "pieChange");
```

Die Funktion "pieChange" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function pieChange(param)
{console.log(param);}");
```

### 2.3.157 chart.event.pieselectchanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	event
---------------------------	-------

Ereignis(Event), das ausgegeben wird, nachdem das Kreisdiagramm die Statusänderungen ausgewählt hat. Es wird ausgelöst, wenn der Benutzer zur Auswahl klickt.

```
dashlet.addOption("chart.event.pieselectchanged", "pieChange");
```

Die Funktion "*pieChange*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function pieChange(param)
{console.log(param);}");
```

### 2.3.158 chart.event.rendered

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Wird ausgelöst, wenn ein Frame gerendert wird. Beachten Sie, dass das gerenderte Ereignis nicht anzeigt, dass die Animation beendet oder das progressive Rendern beendet ist.

```
dashlet.addOption("chart.event.rendered", "renderedFunction");
```

Die Funktion "*renderedFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function renderedFunction(param)
{console.log(param);}");
```

### 2.3.159 chart.event.restore

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Setzt das Optionsereignis zurück.

```
dashlet.addOption("chart.event.rendered", "restoreFunction");
```

Die Funktion *"restoreFunction"* sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function restoreFunction(param)
{console.log(param);}");
```

### 2.3.160 chart.event.timelinechanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Das Ereignis (Event), das nach dem Zeitpunkt in der Zeitleiste ausgegeben wird, wird geändert.

```
dashlet.addOption("chart.event.timelinechanged", "timelineFunction");
```

Die Funktion *"timelineFunction"* sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function timelineFunction(param)
{console.log(param);}");
```

### 2.3.161 chart.event.timelineplaychanged

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	event

Wechselergebnis des Wiedergabestatus in der Timeline.

```
dashlet.addOption("chart.event.timelineplaychanged", "timelineFunction");
```

Die Funktion "*timelineFunction*" sollte später definiert und im Dashboard gerendert werden

```
dashlet.addOption("chart.custom.function", "function timelineFunction(param)
{console.log(param);}");
```

### 2.3.162 chart.series.map.showLegendSymbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.addOption("chart.series.map.showLegendSymbol",
"false");
```

### 2.3.163 chart.series.map.dataRangeHoverLink

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt">"chart.series.map.</span><span
style="color:#2a00ff; font-size:10.0pt">"dataRangeHoverLink</span><span
style="color:#2a00ff; font-size:10.0pt">"</span><span style="color:black; font-size:
10.0pt">, </span><span style="color:#2a00ff; font-size:10.0pt">"false"</span><span
style="color:black; font-size:10.0pt">);</span>
```

### 2.3.164 chart.series.center

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet. <span style="color:black; font-size:10.0pt">(</span><span
style="color:#2a00ff; font-size:10.0pt">"chart.series.center"</span><span
style="color:black; font-size:10.0pt">, </span><span style="color:#2a00ff; font-size:
10.0pt">"[55.2, 10.5]"</span><span style="color:black; font-size:10.0pt">); </
span><span style="color:#3f7f5f; font-size:10.0pt">// latitude/ longitude </span>
```

### 2.3.165 chart.series.zoom

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption</span><span
style="font-size:10.0pt"><span style="color:#3f7f5f">("chart.series.zoom", "1");</
span></span>
```

### 2.3.166 chart.series.map.hoverable

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt"></span><span style="color:#2a00ff; font-
size:10.0pt">chart.series.map.hoverable</span><span style="color:#2a00ff; font-size:
10.0pt">"</span><span style="color:black; font-size:10.0pt">, </span><span
style="color:#2a00ff; font-size:10.0pt">"false"</span><span style="color:black; font-
size:10.0pt">);</span>
```

### 2.3.167 chart.series.map.mapLocation.x

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt">"</span><span style="color:#2a00ff; font-
size:10.0pt">chart.series.map.mapLocation.x</span><span style="color:#2a00ff; font-
size:10.0pt">"</span><span style="color:black; font-size:10.0pt">, </span><span
style="color:#2a00ff; font-size:10.0pt">"100"</span><span style="color:black; font-
size:10.0pt">);</span>
```

### 2.3.168 chart.series.map.mapLocation.y

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt">"</span><span style="color:#2a00ff; font-
size:10.0pt">chart.series.map.mapLocation.y</span><span style="color:#2a00ff; font-
size:10.0pt">"</span><span style="color:black; font-size:10.0pt">, </span><span
style="color:#2a00ff; font-size:10.0pt">"100"</span><span style="color:black; font-
size:10.0pt">);</span>
```

### 2.3.169 chart.series.map.mapLocation.width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt">"chart.series.map.mapLocation</span><span
style="color:#2a00ff; font-size:10.0pt">.width</span><span style="color:#2a00ff;
font-size:10.0pt">"</span><span style="color:black; font-size:10.0pt">, </span><span
style="color:#2a00ff; font-size:10.0pt">"10%"</span><span style="color:black; font-
size:10.0pt">);</span>
```

### 2.3.170 chart.series.map.mapLocation.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	karte, map

Wird in GeoMaps benutzt.

```
dashlet.addOption("chart.type", "map"); dashlet.addOption("chart.series.type", "map")
; dashlet.addOption("chart.map.type", "world");
dashlet.<span style="color:black; font-size:10.0pt">addOption(</span><span
style="color:#2a00ff; font-size:10.0pt">"chart.series.map.mapLocation.</span><span
style="color:#2a00ff; font-size:10.0pt">height</span><span style="color:#2a00ff;
font-size:10.0pt">"</span><span style="color:black; font-size:10.0pt">, </span><span
style="color:#2a00ff; font-size:10.0pt">"10%"</span><span style="color:black; font-
size:10.0pt">);</span>
```

### 2.3.171 locale

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #008000; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	locale

Legt das Locale für das Dashlet fest. Wird hauptsächlich in Pivot-Tabellen und KPI-Dashlet verwendet

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("locale","de_DE");</span>
```

### 2.3.172 id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<span style="background-color: #008000; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	locale

Legt die ID für das Dashlet fest.

Beispiel:

```
<span style="color:black; font-size:10.0pt">dashlet.addOption("id" , "chart1000");</span>
```

### 2.3.173 bar.n[].component.n[].select.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.9
<b>Schlagworte / Tags</b>	singleselect

Mit dieser Option können das Mehrfachauswahlfeld als Einzelauswahlfeld festlegen.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("bar.n[topbar1].component.n[multiselect1].select.type","single");</span>
```

### 2.3.174 box.download.event.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	

Ermöglicht das Anzeigen eines Download Links zu einem generiertem Dokument.

Format ist: "<Custom Event>;<DisplayTitel>"

Beispiel:

```
dashlet.addBoxOption("box.download.event.link", "openJobCreatorDialog([{name:'--__report', value:'\\ServerTest\\a3_landscape_report.rptdesign'},{name:'--__outputFormats', value:'PDF,HTML'}]);event");
```

### 2.3.175 box.download.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Anzeigen eines Download Links zu einem generiertem Dokument.

Format ist: "<DisplayTitel>;<HTML Link>;<Custom Event>;<Target File>"

### 2.3.176 exportChart.dashletId

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	export, exportchart

Definiert die ID(s) der Dashlets / Charts, die in einen BIRT-Report exportiert werden sollen.

Die ID(s) muss eine durch Semikolons getrennte Liste sein.

Beispiel:

```
dashlet.addOption("exportChart.dashletId", "chart1;table1;leaflet1");
dashlet.addOption("exportChart.javascript", "openJobCreatorDialog( [ n" +
" {name:'--__report', value: '/adm/SIT_Print_Image.rptdesign'} n" +
" ,{name:'--__outputFormat', value: 'PDF,HTML'} n" +
" ,{name:'--paramImageMap', value: chart1_img.split(',')[1] } n" +
" ,{name:'--paramImageChart', value: table1_img.split(',')[1] } n" +
" ,{name:'--paramImageTable', value: leaflet1_img.split(',')[1] } n" +
" ]); ");
```

### 2.3.177 exportChart.javascript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	export, exportchart

Definiert die Javascript Funktion, die in servBIRD benutzt wird für die BIRT-Report auszuführen.

Beispiel:

```
dashlet.addOption("exportChart.dashletId", "chart1;table1;leaflet1");
dashlet.addOption("exportChart.javascript", "openJobCreatorDialog( [ n" +
" {name:'--__report', value: '/adm/SIT_Print_Image.rptdesign'} n" +
" ,{name:'--__outputFormat', value: 'PDF,HTML'} n" +
" ,{name:'--paramImageMap', value: chart1_img.split(',')[1] } n" +
" ,{name:'--paramImageChart', value: table1_img.split(',')[1] } n" +
" ,{name:'--paramImageTable', value: leaflet1_img.split(',')[1] } n" +
" ]); ");
```

### 2.3.178 exportChart.dpi

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	export, exportchart

Definiert den dpi (Dots per inch) Wert, der beim Exportieren der Dashlets/Charts in einen BIRT-Report verwendet werden soll.

Beispiel:



```
dashlet.addOption("exportChart.dashletId", "chart1;table1;leaflet1");
dashlet.addOption("exportChart.dpi", "96*1");
dashlet.addOption("exportChart.javascript", "openJobCreatorDialog( [ n" +
" {name:'--__report', value: '/adm/SIT_Print_Image.rptdesign'} n" +
" ,{name:'--__outputFormat', value: 'PDF,HTML'} n" +
" ,{name:'--paramImageMap', value: chart1_img.split(',')[1] } n" +
" ,{name:'--paramImageChart', value: table1_img.split(',')[1] } n" +
" ,{name:'--paramImageTable', value: leaflet1_img.split(',')[1] } n" +
" ]); ");
```

### 2.3.179 chart.series.filter

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.12
<b>Schlagworte / Tags</b>	

Legt die Option zum Filtern der Serie fest.

Beispiel:

```
dashlet3.addOption("chart.series.filter" , "true");
```

### 2.3.180 box.expand.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #ffc107; color: black; padding: 2px;">IMPLEMENTED</span>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Legt die Sichtbarkeit der Schaltfläche 'Expand' für die Box fest.

### 2.3.181 box.expand.hide.ids

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Wird mit der Option "box.expand.show" verwendet. In dieser Option wird eine durch Semikolon getrennte Liste der Dashlet-IDs angegeben, die beim Expandieren des Diagramms ausgeblendet werden sollen.

Beispiel:

```
dashlet.addOption("box.expand.hide.ids", "leaflet1"); //list of dashlets that are to be hidden (semicolon seperated)
```

### 2.3.182 modal.box

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	popup

Definiert die Dashlet-Box als Popup-Box. Soll beispielsweise das DataTable-Dashlet als Popup-Fenster angezeigt werden, wird diese Option für das DataTable-Dashlet definiert.

Beispiel:

```
dashlet.addOption("modal.box", "true");
```

### 2.3.183 box.modal.size

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	box-size, popup

Default-Wert: lg

Definiert die Größe der Dashlet-Box, wenn sie als Popup-Box deklariert ist.

Beispiel:

```
dashlet.addOption("box.modal.size","xl"); // die gröÙe der Drilldown ist dann gröÙ
als die Bildschirm Fenster
```

Wird mit der Option verwendet :

```
dashlet.addOption("modal.box", "true");
```

### Erlaubte Werte

- xl
- sm
- lg
- md
- xs

## 2.4 Dashlet Engine KPI

### 2.4.1 kpi.box.count

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	kpi

Definiert, wie viele Boxen generiert werden sollen.

Diese lassen sich dann über den Index `i[]` Operator ansprechen.

## 2.4.2 `kpi.box.i[].type`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	percentbox, progress

Akzeptierte Werte: "percentbox", "updownbox", "trendbox"

## 2.4.3 `kpi.box.i[].value`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der anzuzeigende Wert innerhalb der KPI Box.

## 2.4.4 `kpi.box.i[].valueseries`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Komma separierte Werte um mehrere anzuzeigen (z.B. bei stacked Progress Bar oder Mini Chart).

Beispiel:

```
dashlet.addOption("kpi.box.i[0].valueseries","17.2,43.5,20.8");
```

## 2.4.5 kpi.box.i[].trendvalue

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeigt einen Veränderungswert (Trend) an.

Beispiel: "17.2"

Achtung: Kann auch String sein!

## 2.4.6 kpi.box.i[].valueformat

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	format, formatierung

Kann der Wert nach einem Pattern formatiert werden.

Beispiel:

```
dashlet.addOption("kpi.box.i[4].valueformat","#0.0%");
```

## 2.4.7 kpi.box.i[].footer

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	fußzeile

Frei definierbarer Text als Anmerkung.

## 2.4.8 kpi.box.i[[]].style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	css, styling

Inline CSS Style

Beispiel:

```
dashlet.addOption("kpi.box.i[0].style","background-color: yellow;");
```

## 2.4.9 kpi.box.i[[]].progressbar.background.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	formatierung, hintergrund

Möglichkeit zum stylen des Fortschrittbalkens.

```
dashlet.addOption("kpi.box.i[0].progressbar.background.style","background-color: #c0ffc0; height: 50px; margin-top: 10px; padding-top: 6px;");
```

## 2.4.10 kpi.box.i[].progressbar.style.{x}

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	fortschrittsbalken, styling

Möglichkeit zum stylen eines einzelnen Elements im gestapelten Fortschrittbalken.

x = Index

Beispiel:

```
dashlet.addOption("kpi.box.i[0].progressbar.style.<span style="color:#ff0000">0</span>","background-color: green; height: 30px;");
dashlet.addOption("kpi.box.i[0].progressbar.style.<span style="color:#ff0000">1</span>","background-color: yellow; height: 30px; color: black;");
dashlet.addOption("kpi.box.i[0].progressbar.style.<span style="color:#ff0000">2</span>","background-color: red; height: 30px;");
```

## 2.4.11 kpi.box.i[].progressbar.class.{x}

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	css-class, css-klasse

Möglichkeit zum setzen einer CSS Klasse eines einzelnen Elements im gestapelten Fortschrittbalken.

{x} = Index

Beispiel:

```
dashlet.addOption("kpi.box.i[0].progressbar.class.0", "progress-bar-success");
dashlet.addOption("kpi.box.i[0].progressbar.class.1", "progress-bar-warning");
dashlet.addOption("kpi.box.i[0].progressbar.class.2", "progress-bar-danger");
```

## 2.4.12 kpi.box.i[].progressbar.tooltip.{x}.show

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einen Tooltip über dem jeweiligen Element des gestapelten Fortschrittbalkens platzieren.

Beispiel:

```
dashlet.addOption("kpi.box.i[0].progressbar.tooltip.0.show", "show");
```

## 2.4.13 kpi.box.i[].progressbar.tooltip.{x}.placement

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	ausrichtung, platzierung

Setzt die Ausrichtung des Tooltips an dem Fortschrittsbalken.

Akzeptierte Werte: top, right, bottom, left

Beispiel:

```
dashlet.addOption("kpi.box.i[0].progressbar.tooltip.0.placement", "bottom");
```



### 2.4.14 kpi.box.i[].progressbar.tooltip.{x}.html

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Inneres HTML im Tooltip des Fortschrittbalkens zulassen.

### 2.4.15 kpi.box.i[].progressbar.tooltip.{x}.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt des Tooltips

### 2.4.16 kpi.box.i[].progressbar.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt des Fortschrittbalkens

### 2.4.17 kpi.box.i[].progressbar.name.{x}

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Inhalt des Fortschrittbalkens

### 2.4.18 kpi.box.i[].footer.visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	fußzeile

Legt fest, ob die Fußzeile angezeigt werden soll oder nicht.

### 2.4.19 kpi.box.i[].footer.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	fußzeile, styling

Möglichkeit zum stylen der Fußzeile

Beispiel:

```
dashlet.addOption("kpi.box.i[0].footer.style","border-top: solid gray 2px;");
```

## 2.4.20 kpi.box.i[].footer.text.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	fußzeile, styling

Möglichkeit zum stylen des Textes der Fußzeile

Beispiel:

```
dashlet.addOption("kpi.box.i[0].footer.text.style","color: green; font-size: 1.5em; text-align: center;");
```

## 2.4.21 kpi.box.i[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	title

Frei als Titel definierbarer Text.

## 2.4.22 kpi.box.i[].title.visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	title

Legt fest, ob die Titelzeile angezeigt werden soll oder nicht.

### 2.4.23 kpi.box.i[].title.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling, title

Möglichkeit zum stylen des Titels

Beispiel:

```
dashlet.addOption("kpi.box.i[0].title.style","border-top: solid gray 2px;");
```

### 2.4.24 kpi.box.i[].title.text.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling, title

Möglichkeit zum stylen des Textes der Titelzeile

Beispiel:

```
dashlet.addOption("kpi.box.i[0].title.text.style","color: green; font-size: 1.5em; text-align: center;");
```

## 2.4.25 kpi.box.i[].value.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling, value

Möglichkeit zum stylen des Wertes

Beispiel:

```
dashlet.addOption("kpi.box.i[0].value.style","color: white; font-size: 1.9em; text-align: left; margin-bottom: -45px; margin-left: 20px;");
```

## 2.4.26 kpi.box.content.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	content, styling

Möglichkeit zum stylen des Inhalts

Beispiel:

```
dashlet.addOption("kpi.box.content.style","height: 120px;");
```

## 2.4.27 kpi.box.i[].value.onclick

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	onclick-function, value

Die Onclick-Funktion für den Wert in der KPI-Box.

Beispiel:

```
dashlet.addOption("kpi.box.i[1].value.onclick","console.log('Click Percent 32');alert('Click Percent 32');");
```

## 2.4.28 kpi.box.i[].echarts.inline.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Anzeige eines ECharts in einer KPI-Box. Die ID des ECharts sollte als Wert für diese Option angegeben werden.

Beispiel:

```
// Wichtige Optionen KPI-Box
dashletKPI.addOption("kpi.box.i[0].echarts.inline.id","chart3");
dashletKPI.addOption("kpi.box.i[0].chart.height","200px");

// Wichtige Optionen EChart
dashletChart.setId("chart3");
dashletChart.addOption("chart.inline.mode", "true");
```

## 2.4.29 kpi.box.i[].chart.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Anzeige eines ECharts in einer KPI-Box. Die Höhe des ECharts sollte als Wert für diese Option angegeben werden.

Beispiel:

```
// Wichtige Optionen KPI-Box
dashletKPI.addOption("kpi.box.i[0].echarts.inline.id","chart3");
dashletKPI.addOption("kpi.box.i[0].chart.height","200px");

// Wichtige Optionen EChart
dashletChart.setId("chart3");
dashletChart.addOption("chart.inline.mode", "true");
```

## 2.4.30 kpi.box.i[].trend.chart.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein von 'line' (default), 'bar', 'tristate', 'discrete', 'bullet', 'pie' oder 'box'

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.type", "bar");
```

### 2.4.31 kpi.box.i[].trend.chart.width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Diagramms - default 'auto' - kann eine beliebige gültige CSS-Breite sein - 1.5em, 20px usw. (eine Zahl ohne Einheitenspezifizierer macht nicht das, was Sie wollen). Diese Option bewirkt nichts in Bar und TriState Charts (siehe barWidth)

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.width", "'150'");
dashlet.addOption("kpi.box.i[1].trend.chart.width", "'100%'");
```

### 2.4.32 kpi.box.i[].trend.chart.drawNormalOnTop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Standardmäßig wird der normale Bereich hinter dem Füllbereich gezeichnet. Wenn Sie diese Option auf true setzen, wird sie über den Füllbereich gezogen

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.drawNormalOnTop", "false");
```

### 2.4.33 kpi.box.i[].trend.chart.valueSpots

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Gibt an, welchen Punkten mit welcher Farbe gezeichnet werden sollen. Akzeptiert einen Bereich. Verwenden Sie beispielsweise {'49': 'green', '50': 'red'}, um grüne Punkte auf allen Werten unter 50 und rote auf höheren Werten zu rendern.

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.valueSpots", "{':  
34.9': 'green', '35.0': 'red'}");
```

### 2.4.34 kpi.box.i[].trend.chart.lineColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Linie in einem Line oder Discrete Charts als CSS-Wert

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.lineColor","#0000FF");
```

### 2.4.35 kpi.box.i[].trend.chart.chartRangeMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mindestwert in Y-Bereich des Diagramms. Der Standardwert ist der angegebene Mindestwert

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.chartRangeMin","1");
```

### 2.4.36 kpi.box.i[].trend.chart.chartRangeMax

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Maximalwert in Y-Bereich des Diagramms. Der Standardwert ist der angegebeneMaximalwert

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.chartRangeMax","10");
```

### 2.4.37 kpi.box.i[].trend.chart.defaultPixelsPerValue

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Standardmäßig verwendet die Bibliothek 3 Pixel als Breite für jeden Datenpunkt im Diagramm. Verwenden eines einzelnen Pixels zur Maximierung der Informationsdichte.

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.defaultPixelsPerValue","1");
```

### 2.4.38 kpi.box.i[].trend.chart.spotColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe des ausgewählten Markers. Setzen Sie auf false oder eine leere Zeichenfolge, um sie auszublenden

Beispiel:

```
ashlet.addOption("kpi.box.i[1].trend.chart.spotColor","'#FF0000'");
```

### 2.4.39 kpi.box.i[].trend.chart.minSpotColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die CSS-Farbe des Markers mit dem minimalen Wert. 'False' oder eine leere Zeichenfolge, um auszublenden  
Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.minSpotColor","#00FFFF");
```

#### 2.4.40 kpi.box.i[].trend.chart.maxSpotColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die CSS-Farbe des Markers mit dem maximalen Wert. 'False' oder eine leere Zeichenfolge, um auszublenden  
Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.maxSpotColor","#00FFFF");
```

#### 2.4.41 kpi.box.i[].trend.chart.spotRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

In Line: Radius von allen Markers in Pixels. Default 1.5 (Integer)

In Box: Radius der Ausreißerkreise in Pixel

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.spotRadius","3");
```

## 2.4.42 kpi.box.i[].trend.chart.highlightSpotColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Markerfarbe für den Wert, der angezeigt wird, wenn Sie mit der Maus darüber fahren. Zum Deaktivieren auf null setzen. Der Standardwert ist #f5f

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.highlightSpotColor","'#808080'");
```

## 2.4.43 kpi.box.i[].trend.chart.highlightLineColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Gibt eine Farbe für die vertikale Linie an, die beim Bewegen der Maus durch einen Wert angezeigt wird. Zum Deaktivieren auf null setzen. Der Standardwert ist #f22

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.highlightSpotColor","#808080");
```

## 2.4.44 kpi.box.i[].trend.chart.lineWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienbreite In pixels (default: 1) - Integer

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.lineWidth","3");
```

## 2.4.45 kpi.box.i[].trend.chart.normalRangeMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>IMPLEMENTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

(siehe auch normalRangeMax)

Geben Sie Schwellenwerte an, zwischen denen ein Balken gezeichnet werden soll, um den "normalen" oder erwarteten Wertebereich anzugeben. Zum Beispiel das Grün für einen normalen Betriebstemperaturbereich

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.normalRangeMin","40.0");
```

## 2.4.46 kpi.box.i[].trend.chart.normalRangeMax

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

(siehe auch *normalRangeMin*)

Geben Sie Schwellenwerte an, zwischen denen ein Balken gezeichnet werden soll, um den "normalen" oder erwarteten Wertebereich anzugeben. Zum Beispiel das Grün für einen normalen Betriebstemperaturbereich

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.normalRangeMin","40.0");
```

## 2.4.47 kpi.box.i[].trend.chart.chartRangeMaxX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Maximale Wert für den X-Bereich

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.chartRangeMaxX","1");
```

## 2.4.48 kpi.box.i[].trend.chart.zeroAxis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zentriert die y-Achse auf 0, wenn true (default)

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.zeroAxis","true");
```

## 2.4.49 kpi.box.i[].trend.chart.xvalues

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Übergeben Sie x-Werte separat:

```
$ ('# linechart'). Sparkline ([3,4,3], {xvalues: [1,2.7,4.8]});
```

## 2.4.50 kpi.box.i[].trend.chart.tooltipOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	



Entfernung in Pixel vom Mauszeiger bis zum Tooltip auf der X-Achse

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.tooltipOffsetX","10");
```

### 2.4.51 kpi.box.i[].trend.chart.barSpacing

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen den Balken in Pixel (Integer)

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.barSpacing","10");
```

### 2.4.52 kpi.box.i[].trend.chart.barColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für positive Werte

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.barColor","'#FF00FF'");
```

### 2.4.53 kpi.box.i[].trend.chart.negBarColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für negative Werte

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.negBarColor", "'#FFFF00'");
```

### 2.4.54 kpi.box.i[].trend.chart.zeroColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für 0-Werte

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.zeroColor", "'#0000FF'");
```

### 2.4.55 kpi.box.i[].trend.chart.nullColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für null-Werte. Standardmäßig werden Nullwerte vollständig weggelassen. Mit der Option wird eine dünne Markierung für den Eintrag hinzugefügt. Dies kann nützlich sein, wenn Ihr Diagramm ziemlich dünn ist. Versuchen Sie eine unauffällig Farbe (z.B. hellgrau)

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.nullColor","#000000");
```

## 2.4.56 kpi.box.i[].trend.chart.colorMap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Eine Bereichskarte zum Zuordnen bestimmter Werte zu ausgewählten Farben. Z.B. alle Werte von -2 werden gelb angezeigt: colorMap: {'-2': '#ff0'}.

Ab Version 1.5 können Sie hier anstelle einer Zuordnung auch ein Array von Werten übergeben, um eine Farbe für jeden einzelnen Balken anzugeben. Wenn Ihr Diagramm beispielsweise drei Werte 1,3,1 hat, können Sie colorMap = ['red','green','blue'] festlegen.

Ab Version 2.0 werden Objekte automatisch in Bereichskarten konvertiert, sodass Sie Farben für einen Wertebereich angeben können

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.colorMap",["red','green','blue']);
```

## 2.4.57 kpi.box.i[].trend.chart.stackedBarColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Eine Reihe von Farben für gestapelte Balkendiagramme. Die erste Reihe verwendet den ersten Wert im Array, die zweite Reihe den zweiten usw.

### 2.4.58 `kpi.box.i[].trend.chart.posBarColor`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für positive (Gewinn-) Werte

### 2.4.59 `kpi.box.i[].trend.chart.zeroBarColor`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

S-Farbe für Nullwerte (Zeichnen)

### 2.4.60 `kpi.box.i[].trend.chart.lineHeight`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box

<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe jeder Linie in Pixel - Standardmäßig 30% der Diagrammhöhe.

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.lineHeight","20");
```

### 2.4.61 kpi.box.i[].trend.chart.thresholdValue

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Werte, die unter diesem Wert liegen, werden mit schwellenwertfarbe anstelle von linienfarbe gezeichnet. Default: 0

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.thresholdValue","40");
```

### 2.4.62 kpi.box.i[].trend.chart.thresholdColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	TESTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe zum Benutzen in Kombination mit thresholdValue (Schwellenwert)

Default: Null

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.thresholdColor","#FF00FF");
```

### 2.4.63 kpi.box.i[].trend.chart.targetColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die CSS-Farbe des vertikalen Zielmarkers

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.targetColor","#00AA00");
```

### 2.4.64 kpi.box.i[].trend.chart.targetWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Markerbreite in pixels (default: 3) - Integer

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.targetWidth","10");
```

## 2.4.65 kpi.box.i[].trend.chart.performanceColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Farbe des Leistungsmaßes horizontaler Balken. Defaults to: #33f

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart. performanceColor", "#FF00FF");
```

## 2.4.66 kpi.box.i[].trend.chart.sliceColors

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Array von CSS-Farben, die für Tortenstücke verwendet werden sollen

Beispiel:

```
dashlet.addOption("kpi.box.i[2].trend.chart.sliceColors",
"['#00AA00', '#00FF00', '#0000FF']");
```

## 2.4.67 kpi.box.i[].trend.chart.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Winkel in Grad, um die erste Schicht zu versetzen (zwischen -90 oder +90)

Beispiel:

```
dashlet.addOption("kpi.box.i[2].trend.chart.offset","20");
```

## 2.4.68 kpi.box.i[].trend.chart.borderWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Rahmens um das Kreisdiagramm in Pixel.  
Default ist 0 (kein Rand)

Beispiel:

```
dashlet.addOption("kpi.box.i[2].trend.chart.borderWidth","1");
```

## 2.4.69 kpi.box.i[].trend.chart.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

CSS-Farbe zum Zeichnen des Kreisrahmens.

Default ist # 000

Beispiel:

```
dashlet.addOption("kpi.box.i[2].trend.chart.borderColor","#00FF00");
```

## 2.4.70 kpi.box.i[].trend.chart.raw

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Standardmäßig werden die Punkte aus dem Eingabewertarray berechnet. Setzen Sie dies auf true, um die vorberechneten Punkte in der Wertekonfiguration zu übergeben. Default: false

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.raw","true");
```

## 2.4.71 kpi.box.i[].trend.chart.showOutliers

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Sehe auch .outlierIQR

Wenn true (default), werden Ausreißer (Werte > 1,5x IQR) mit Kreisen markiert. Whisker werden bei Q1 und Q3 platziert anstelle des kleinsten und größten Wertes

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.showOutliers","false");
```

## 2.4.72 kpi.box.i[].trend.chart.outlierIQR

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Multiplikator für den Bereich zwischen Quartilen, der zur Berechnung von Ausreißer verwendet wird. Der Standardwert ist 1,5

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.outlierIQR","0.5");
```

## 2.4.73 kpi.box.i[].trend.chart.boxLineColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-farbe der Linie, mit der die Box umrissen wird

Beispiel:

```
dashlet.addOption("kpi.box.i[2].trend.chart.boxLineColor","'#00FF00'");
```

## 2.4.74 kpi.box.i[].trend.chart.boxFillColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Füllfarbe für das Box

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.boxFillColor","#FFFF00");
```

## 2.4.75 kpi.box.i[].trend.chart.whiskerColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Füllfarbe für das Box

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.whiskerColor","#000000");
```

## 2.4.76 kpi.box.i[].trend.chart.outlierFillColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für Ausreißerkreise

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.outlierFillColor","#FFFF00");
```

## 2.4.77 kpi.box.i[].trend.chart.minValue

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn Min- und Max-Wert eingestellt sind, ist der Maßstab des Diagramms festgelegt. Standardmäßig werden minValue und maxValue aus den angegebenen Werten abgeleitet

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.minValue","10");
```

## 2.4.78 kpi.box.i[].trend.chart.maxValue

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn Min- und Max-Wert eingestellt sind, ist der Maßstab des Diagramms festgelegt. Standardmäßig werden `minValue` und `maxValue` aus den angegebenen Werten abgeleitet

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.maxValue","10");
```

## 2.4.79 kpi.box.i[].trend.chart.medianColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe der Medianlinie.

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.medianColor","'#00FF00'");
```

## 2.4.80 kpi.box.i[].trend.chart.barSpacing

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn ein Wert festgelegt ist, wird an dieser Stelle ein kleines Fadenkreuz gezeichnet, um einen Zielwert darzustellen

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.target","55");
```

### 2.4.81 kpi.box.i[].trend.chart.disableInteraction

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

True - um die gesamte Sparkline-Interaktivität zu deaktivieren. Dadurch verhält sich das Plugin ähnlich wie in 1.x.  
Default false

### 2.4.82 kpi.box.i[].trend.chart.disableTooltips

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Bar
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

True - um Mouseover-Tooltips zu deaktivieren.

Default false

### 2.4.83 kpi.box.i[].trend.chart.highlightLighten

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Ein Wert beim Bewegen der Maus wird aufgehellt oder abgedunkelt.  
1,5- Wert wird um 50% aufgehellt; 0,5 - Wert wird um 50% abgedunkelt.  
Der Standardwert ist 1.4

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.highlightLighten","1.5");
```

## 2.4.84 kpi.box.i[].trend.chart.tooltipOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Entfernung in Pixel vom Mauszeiger bis zum Tooltip auf der Y-Achse

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.tooltipOffsetY","10");
```

## 2.4.85 kpi.box.i[].trend.chart.tooltipFormatter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Callback-Javascript-Funktion zum Überschreiben des HTML-Codes für Generieren von QuickInfos. Dem Callback werden Argumente übergeben (sparkline, optionen, felder).

"sparkline" ist das Sparkline-Objekt, das gerendert wird.

"options" ist key:value mapping für Optionen. Verwenden Sie options.get(key, default), um eine Option abzurufen.

"Felder" ist ein Array von Feldern, die für die Sparkline gerendert werden sollen. Dies ist ein Array mit einem einzelnen Element, es sei denn, es handelt sich um ein Box-Diagramm.

## 2.4.86 kpi.box.i[].trend.chart.tooltipChartTitle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Wenn angegeben, verwendet der Tooltip die in dieser Einstellung angegebene Zeichenfolge als Titel

## 2.4.87 kpi.box.i[].trend.chart.tooltipFormat

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Eine Formatstring oder ein sprintf-Objekt (oder ein Array davon - für mehrere Einträge), um das Format der QuickInfo zu steuern

## 2.4.88 kpi.box.i[].trend.chart.tooltipPrefix

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Eine Zeichenfolge, die jedem in einem Tooltip angezeigten Feld vorangestellt wird



## 2.4.89 `kpi.box.i[].trend.chart.tooltipSuffix`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Eine Zeichenfolge, die an jedes in einer QuickInfo angezeigte Feld angehängt werden soll

## 2.4.90 `kpi.box.i[].trend.chart.tooltipSkipNull`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Wenn true, wird für Nullwerte kein Tooltip angezeigt  
Der Standardwert ist true

## 2.4.91 `kpi.box.i[].trend.chart.tooltipValueLookups`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Ein Objekt- oder Range-Map zum Zuordnen von Feldwerten zu Tooltip-Zeichenfolgen. Beispielsweise werden -1, 0 und 1 den Zeichenfolgen "Lost", "Draw", "Won" zugeordnet.

## 2.4.92 kpi.box.i[].trend.chart.tooltipFormatFieldlist

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Ein Array von Werten, die angeben, welche Felder in einem Tooltip in welcher Reihenfolge angezeigt werden sollen. Derzeit nur für Box-Plots nützlich. Man kann steuern, welche Felder in welcher Reihenfolge angezeigt werden. Zum Beispiel, um nur den Median, das untere Quartil und das obere Quartil in dieser Reihenfolge anzuzeigen:

```
dashlet.addOption("kpi.box.i[0].trend.chart.tooltipFormatFieldlist",["med', 'lq', 'uq']");
```

## 2.4.93 kpi.box.i[].trend.chart.tooltipFormatFieldlist

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Sieh auch tooltipFormatFieldlist

Gibt an, welcher Schlüssel den Feldnamen enthält, auf den in tooltipFormatFieldlist verwiesen wird. Für Box-Plots sollte dies "Feld" sein.

```
dashlet.addOption("kpi.box.i[0].trend.chart.tooltipFormatFieldlistKey", "field");
```

## 2.4.94 kpi.box.i[].trend.chart.tooltipContainer

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Gibt das DOM-Element an, in das QuickInfos gerendert werden sollen.  
Der Standardwert ist document.body

#### 2.4.95 kpi.box.i[].trend.chart.tooltipClassname

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Gibt einen CSS-Klassennamen an, der auf QuickInfos angewendet werden soll, um den integrierten Standardstil zu überschreiben.

#### 2.4.96 kpi.box.i[].trend.chart.numberFormatter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Übergeben Sie eine Javascript-Funktion, um zu steuern, wie Zahlen in Tooltips formatiert werden. Der Rückruf erhält eine zu formatierende Nummer und muss eine Zeichenfolge zurückgeben. Standardmäßig werden Zahlen nach westlichen Konventionen formatiert.

### 2.4.97 kpi.box.i[].trend.chart.numberDigitGroupSep

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Zeichen, das für das Gruppentrennzeichen in den Nummern "1.234" für 10n-Zwecke verwendet werden soll.  
Der Standardwert ist das Komma - ","

### 2.4.98 kpi.box.i[].trend.chart.numberDecimalMark

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Ein Zeichen, das für den Dezimalpunkt in Zahlen für 10n Zwecke verwendet werden soll.  
Der Standardwert ist der Punkt - "."

### 2.4.99 kpi.box.i[].trend.chart.numberDigitGroupCount

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.6
<b>Schlagworte / Tags</b>	

Anzahl der Ziffern zwischen dem Gruppentrenner in Zahlen für 10n Zwecke.  
Der Standardwert ist 3.

### 2.4.100 kpi.box.i[*n*].icon.up

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol

Ermöglicht das Setzen eines Symbols für 'UP' (positive Werte).

Siehe auch: <https://fontawesome.com/icons>

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[2].icon.up",'fa fa-arrow-circle-o-up');</span>
```

### 2.4.101 kpi.box.i[*n*].icon.down

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol


Ermöglicht das Setzen eines Symbols für 'DOWN' (negative Werte).

Siehe auch: <https://fontawesome.com/icons>

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.down",'fa fa-arrow-circle-o-down');</span>
```

## 2.4.102 kpi.box.i[].icon.down.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol, symbol-style


Ermöglicht das Festlegen eines Stils für das Symbol 'DOWN' (negative Werte).

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.down",'fa fa-arrow-circle-o-down');</span>

<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.down.style","color: red;");</span>
```

## 2.4.103 kpi.box.i[].icon.up.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol, symbol-style

Ermöglicht das Festlegen eines Stils für das Symbol 'UP' (positive Werte).

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.up",'fa fa-arrow-circle-o-up');</span>

<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.up.style","color: green;");</span>
```

### 2.4.104 kpi.box.i[].label.down

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol-label

Ermöglicht das Festlegen einer Beschriftung für das Symbol 'DOWN' (negative Werte).

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[2].label.down","Crash");</span>
```

### 2.4.105 kpi.box.i[].label.up

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol-label

Ermöglicht das Festlegen einer Beschriftung für das Symbol 'UP' (positive Werte).

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[2].label.up","Gut");</span>
```

### 2.4.106 kpi.box.i[].value.up.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	value-style

Allows setting a style for the value in case of 'UP' (positive values) values.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.up",'fa fa-arrow-circle-o-up');</span>
```

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].value.up.style","color : green;");</span>
```

#### 2.4.107 kpi.box.i[].value.down.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	value-style

Allows setting a style for the value in case of 'DOWN' (negative values) values.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.down",'fa fa-arrow-circle-o-down');</span>
```

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].value.down.style","color: red;");</span>
```

#### 2.4.108 kpi.box.i[].trendvalue.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI



<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legt den Stil für einen Änderungswert (Trend) fest.

### 2.4.109 kpi.box.i[].trendvalueformat

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	format, formatierung

Kann der Wert nach einem Pattern formatiert werden.

Beispiel:

```
dashlet.addOption("kpi.box.i[4].trendvalueformat", "#0.0%");
```

### 2.4.110 kpi.box.responsive

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	
<b>Schlagworte / Tags</b>	

Legt die Option für die Reaktionsfähigkeit der KPI-Box fest. Die erste Wert ist for die 'Large' Display units, die zweite ist for 'Medium' Display units, die dritte für 'Small' display units und die vierte für 'Extra-Small' Display units.

Beispiel:

```
dashlet.addOption("kpi.box.responsive","2-3-4-12");
```

In das HTML ist die KPI-class als folgendes gerendert

```
class="col-lg-2 col-md-3 col-sm-4 col-xs-12"
```

### 2.4.111 kpi.box.i[].icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol

Ermöglicht das Setzen eines Symbols

Siehe auch: <https://fontawesome.com/icons>

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon",'</span>fa fa-sitemap<span style="font-size:10.0pt">');</span>
```

### 2.4.112 kpi.box.i[].trend.chart.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, die Höhe des Trenddiagramms in der KPI-Box festzulegen

Beispiel:

```
dashlet.addOption("kpi.box.i[0].trend.chart.height","20");
```

### 2.4.113 kpi.box.i[].trend.chart.fillColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

CSS-Farbe für positive Werte

Beispiel:

```
dashlet.addOption("kpi.box.i[1].trend.chart.fillColor","#FF00FF");
```

### 2.4.114 box.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	styling

Legt den Stil der KPI-Box fest

Beispiel:

```
dashlet.addOption("box.style","background-color:white;"); // set the style of the KPI
Box
```

### 2.4.115 kpi.box.i[].trendvalue.visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI

<b>Komponente</b>	KPI Box
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	

Legt den Visibility für einen Änderungswert (Trend) fest.

Beispiel:

```
dashlet.addOption("kpi.box.trendvalue.visible","false");
```

### 2.4.116 kpi.box.i[].icon.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	symbol, symbol-style

Ermöglicht das Festlegen eines Stils für das Symbol Trend.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon",'fa fa-arrow-circle-o-down');</span>
```

```
<span style="font-size:10.0pt">dashlet.addOption("kpi.box.i[3].icon.style","color:red;");</span>
```

### 2.4.117 kpi.box.i[].icon.visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	KPI
<b>Komponente</b>	Box

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.10
<b>Schlagworte / Tags</b>	symbol

Ermöglicht das Festlegen eine Visibility für das Symbol Trend.

Beispiel:

```
dashlet.addOption("kpi.box.icon.visible","false");
```

## 2.5 Dashlet Engine Special Dashlet

### 2.5.1 parameter.layout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Format: flow

Zeigt das Layout der im Report definierten Parameter an.

Wurde das Format 'flow' oder 'auto' gewählt und die Zeichenanzahl ist größer als 240, so wird automatisch das 'columnar'-Format verwendet.

Erlaubte Werte

- auto
- flow
- columnar

### 2.5.2 parameter.toggleVisibility

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : false

Zeigt an, ob die Parameter immer angezeigt werden sollen oder über einen Knopf zusammenklappbar sein sollen.

Steht der Wert auf 'false', so wird der Knopf zum Zusammenklappen nicht angezeigt.

### 2.5.3 parameter.toggleVisibility.defaultState

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : expanded

Steht der toggleVisibility-Parameter auf true, so kann der Zustand des Knopfes zum Zusammenklappen durch diesen Parameter definiert werden.

#### Erlaubte Werte

- expanded
- collapsed

### 2.5.4 parameter.showHiddenParameters

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default-Wert : false

Zeigt an, ob verborgene Parameter angezeigt werden sollen oder nicht. Zu Debugzwecken verwendet.

## 2.5.5 parameter.multiValueSeperator

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default-Wert : commaseperated

Spezifiziert den Separator der verwendet werden sollte, wenn der Parameter über Mehrfachauswahl verfügt.

Erlaubte Werte

- commaseperated
- linebreak

## 2.5.6 parameter.helpTextAsTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : false

Spezifiziert, ob der 'HelpText', der für den Parameter definiert wurde, im Parameter-Dashlet angezeigt werden soll oder nicht.

## 2.5.7 parameter.showPasswordClearText

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : false

Zeigt an, ob die Werte von Parametern die den Typ 'password' haben angezeigt werden sollen oder nicht. Zu Debugzwecken verwendet.

### 2.5.8 parameter.param.n[].show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : true

Ein Parameter kann durch diese Option angezeigt oder verborgen werden. Das 'ParameterName'-Feld des Parameters sollte als Name benutzt werden.

### 2.5.9 parameter.param.n[].labelText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Die Beschriftung eines Parameters kann durch diese Option modifiziert werden. Das 'ParameterName'-Feld des Parameters sollte als Name benutzt werden.



### 2.5.10 parameter.param.n[].helpText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	parameter dashlet

Der Hilfstext eines Parameters kann durch diese Option modifiziert werden. Das 'ParameterName'-Feld des Parameters sollte als Name benutzt werden.

### 2.5.11 parameter.showTextOutputParameters

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Special Dashlet
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.15
<b>Schlagworte / Tags</b>	parameter dashlet

Default-Wert : false

Gibt an, ob die Parameter mit der Benutzereigenschaft displayText = TextOutput angezeigt werden sollen oder nicht.

## 3 DataTables [Options]

<b>Name</b>	DataTables
<b>Version</b>	1.10
<b>Hersteller</b>	SpryMedia Ltd.
<b>Webseite</b>	<a href="https://datatables.net">https://datatables.net</a>

### 3.1 DataTables Buttons

#### 3.1.1 table.buttons.buttons

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Buttons
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Man beachte: Dieses Attribut setzt die Buttons-Erweiterung für DataTables voraus.

Beispiel:

```
dashlet.addOption("table.buttons.buttons" , "[ 'csv', 'excel' ]");
```

Das buttons-Array definiert die Knöpfe, die im Dokument dem Endnutzer angezeigt werden. Jedes Element kann einen der folgenden Typen haben:

- String - Der Name eines Knopftyps der eingebaut ist oder durch ein Plugin hinzugefügt wurde. Dies ist das Äquivalent zur Verwendung von { extend: ... }.
- Objekt - Ein eigener Knopf oder eine angepasste Form eines existierenden Knopftyps. Die extend-Eigenschaft (siehe buttons.buttons.extend) kann definiert werden, um Buttons zu zeigen, auf welchem Knopftyp der Knopf basieren soll und welche Eigenschaften angepasst werden sollen. Andernfalls, wenn extend nicht gesetzt ist, kann ein Knopf über die Optionen text und action definiert werden. Buttons hat nur wenige eingebaute Konfigurationsoptionen für jeden Knopf, aber die jeweiligen Knopftypen können zusätzliche Optionen zur Verfügung stellen. Siehe die Dokumentation der einzelnen Knopftypen für Informationen über die verfügbaren Optionen. Die standardmäßigen Optionen sind:
  - buttons.buttons.action - Funktion die nach aktivierung ausgeführt wird
  - buttons.buttons.className - Klassenname des Knopfs
  - buttons.buttons.enabled - Initialer Aktivierungsstatus
  - buttons.buttons.extend - Erweitertes Basisobjekt
  - buttons.buttons.init - Callback-Funktion der Initialisierung
  - buttons.buttons.key - Konfiguration der Schlüsselaktivierung

- `buttons.buttons.name` - Knopfname für die Verwendung in Selektoren
- `buttons.buttons.text` - Sichtbarer Text
- Funktion - Eine Funktion die bei der Erstellung der Knöpfe ausgeführt wird. Die Funktion hat nur ein Argument, die DataTables API-Instanz der übergeordneten Tabelle, und der Rückgabewert sollte eine der oberen Optionen oder ein Array aus diesen Optionen sein.

### 3.1.2 Typ

#### array

Dieses Array definiert eine Liste an Knöpfen, die in der Knopfsammlung angezeigt werden sollen. Die Elemente des Arrays können die Typen String, Objekt und Funktion haben. Siehe die obige Dokumentation für Details. Das Array kann mehrere Elemente jeden Typs enthalten.

### 3.1.3 Beispiele

DataTables-Initialisierung: buttons als Array simpler Typen:

```
$('#myTable').DataTable( {  
  buttons: [ 'copy', 'csv', 'print' ]  
} );
```

DataTables-Initialisierung: Zwei Knöpfe, einer davon mit Anpassungen:

```
$('#myTable').DataTable( {  
  buttons: {  
    buttons: [  
      'copy',  
      { extend: 'excel', text: 'Save as Excel' }  
    ]  
  }  
} );
```

DataTables-Initialisierung: Ein komplett eigener Knopf:

```
$('#myTable').DataTable( {  
  buttons: {  
    buttons: [  
      'copy',  
      {  
        text: 'My button',  
        action: function ( dt ) {  
          console.log( 'My custom button!' );  
        }  
      }  
    ]  
  }  
} );
```

Instanz-Initialisierung: Verwendung des Knopf-Arrays:

```
new $.fn.dataTable.Buttons( table, {
  buttons: [
    'copy',
    { extend: 'excel', text: 'Save as Excel' }
  ]
} );
```

## 3.2 DataTables General

### 3.2.1 table.feature.autoWidth

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Automatische Berechnung der Spaltenbreite aktivieren oder deaktivieren. Diese Option kann als Optimierung deaktiviert werden (das Berechnen der Breiten benötigt eine endliche Menge an Zeit), wenn die Breiten der Tabelle über die `columns.width` Option übergeben werden.

### 3.2.2 Beispiel

Intelligente Breitenberechnung deaktivieren:

```
$('#example').dataTable( {
  "autoWidth": false
} );
```

### 3.2.3 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `columns.adjust()`

Events

- `column-sizing`

## Options

- columns.width

## table.feature.deferRender

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Standardmäßig generiert DataTables sofort alle benötigten HTML-Elemente wenn Daten aus einer Ajax- oder Javascript-Datenquelle (ajax Option bzw. data Option) geladen werden. Wenn mit großen Datensets gearbeitet wird, kann diese Operation eine signifikante Menge an Zeit in Anspruch nehmen, vor allem in alten Browsern wie IE6-8. Diese Option erlaubt es DataTables die Knoten (Reihen und Zellen im Tabellenkörper) erst zu generieren, wenn sie für einen draw benötigt werden.

Wird zum Beispiel ein Datenset mit 10.000 Zeilen geladen, aber pro Seite sollen nur 10 Einträge dargestellt werden, so wird DataTables mit aktiviertem deferred Rendering nur 10 Reihen statt 10.000 generieren. Wenn der Nutzer nun blättert oder die Daten sortiert oder filtert, werden automatisch die für den nächsten draw fehlenden Elemente generiert. So wird die Generierungslast der Elemente über die Lebensdauer der Seite verteilt.

Es ist zu beachten, dass wenn die Option aktiviert ist, nicht alle Knoten immer in der Tabelle verfügbar sind, was bei der Verwendung von API-Methoden wie columns().nodes() bedacht werden muss. Unten wird ein Beispiel gegeben, wie man mit einer solchen Situation mit jQuery delegierten Events umgehen kann.

### 3.2.4 Beispiele

Deferred Rendering aktivieren:

```
$('#example').dataTable( {
  "ajax": "sources/arrays.txt",
  "deferRender": true
} );
```

Events mit deferred Rendering:

```

$('#example tbody').on( 'click', 'td', function () {
    alert( 'Angeclickt wurde: '+this.innerHTML );
} );

$('#example').dataTable( {
    "ajax": "sources/arrays.txt",
    "deferRender": true
} );

```

### 3.2.5 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax
- data

table.feature.info

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ist diese Option aktiviert, so wird DataTables Informationen über die Tabelle anzeigen, sowie Informationen über gefilterte Daten, falls diese Aktion durchgeführt wurde. Mit dieser Option kann dieses Verhalten aktiviert und deaktiviert werden.

Es ist anzumerken, dass die Informationen standardmäßig links unter der Tabelle angezeigt werden, aber dies kann über die dom Option und CSS kontrolliert werden.

### 3.2.6 Beispiel

Anzeigen der Informationen deaktivieren:

```

$('#example').dataTable( {
    "info": false
} );

```

### 3.2.7 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.info
- language.infoEmpty
- language.infoFiltered
- infoCallback

#### table.feature.lengthChange

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ist Paginierung aktiviert, so kontrolliert diese Option das Bereitstellen einer Möglichkeit für den Nutzer, die Anzahl der pro Seite angezeigten Einträge zu verändern. Die möglichen Werte für diese Anzahl werden durch die lengthMenu Option kontrolliert.

Es ist anzumerken, dass die Steuerung standardmäßig links über der Tabelle angezeigt wird, dies kann über die dom Option und CSS kontrolliert werden.

Ist diese Option deaktiviert (false), so wird die Steuerung für die Seitengröße nicht angezeigt. Die page.len() API-Methode kann jedoch weiterhin verwendet werden, falls die Seitengröße programmatisch verändert werden soll, und die pageLength Option kann verwendet werden, um die initiale Seitenlänge zu spezifizieren. Die Seitenbildung selbst wird nicht beeinflusst.

Weiterhin wird diese Option automatisch deaktiviert, wenn Paginierung über die paging Option deaktiviert wird, da sie dann keine Relevanz mehr hat.

### 3.2.8 Beispiel

Deaktiviere festlegen der Einträge pro Seite durch den Nutzer:

```
$('#example').dataTable( {
  "lengthChange": false
} );
```

### 3.2.9 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `page.len()`

Options

- `dom`
- `lengthMenu`
- `language.lengthMenu`
- `pageLength`

`table.feature.ordering`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Sortieren nach Spalten aktivieren oder deaktivieren - so einfach ist es! DataTables erlaubt es Nutzern standardmäßig, auf die Kopfzeile jeder Spalte zu klicken, um die Daten nach dieser Spalte zu sortieren. Mit dieser Option kann diese Fähigkeit deaktiviert werden.

Es ist anzumerken, dass die Fähigkeit nach einer Spalte zu sortieren über die `columns.orderable` Option für jede Spalte einzeln deaktiviert werden kann. Die Option `ordering` ist eine globale Option - ist sie deaktiviert, werden gar keine Sortieraktionen mehr von DataTables ausgeführt.

### 3.2.10 Beispiel

Sortierung in der Tabelle deaktivieren:

```
$('#example').dataTable( {
  "ordering": false
} );
```

### 3.2.11 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API



- `order()`
- `column().order()`
- `columns().order()`

## Options

- `columns.orderable`
- `order`
- `orderFixed`
- `orderMulti`
- `orderClasses`

## table.feature.paging

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

DataTables kann die Zeilen einer Tabelle auf individuelle Seiten aufteilen, was eine effiziente Methode ist, eine große Menge Einträge auf kleinem Platz darzustellen. Dem Nutzer werden Kontrollmöglichkeiten zur Verfügung gestellt, um sich andere Daten anzeigen zu lassen. Dieses Feature ist standardmäßig aktiviert, aber soll es deaktiviert sein, kann dies mit dieser Option getan werden.

### 3.2.12 Beispiel

Paginierung deaktivieren:

```
$('#example').dataTable( {
  "paging": false
} );
```

### 3.2.13 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

## API

- `page()`

## Options

- `info`
- `lengthChange`
- `lengthMenu`

- pagingType

### table.feature.processing

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Aktiviert oder deaktiviert das Anzeigen eines 'In Arbeit'-Indikators, wenn die Tabelle bearbeitet wird (z.B. während einer Sortierung). Dies ist nützlich für Tabellen mit großen Datenmengen, bei denen das Sortieren der Daten eine merkliche Zeit in Anspruch nimmt.

### 3.2.14 Beispiel

Anzeigen des Bearbeitungsindikators aktivieren:

```
$('#example').dataTable( {
  "processing": true
} );
```

### 3.2.15 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.processing

### table.feature.scrollX

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Aktiviert horizontales scrollen. Wenn eine Tabelle zu Breit ist, um in ein bestimmtes Layout zu passen, oder eine hohe Anzahl Spalten hat, kann horizontales scrollen aktiviert werden, um die Tabelle in einem horizontal scrollbaren Viewport anzuzeigen.

Diese Eigenschaft kann auf true gesetzt werden, wodurch die Tabelle horizontal scrollen kann, falls nötig (empfohlen), oder auf eine beliebige CSS Einheit, oder auf eine Zahl (welche als Pixelwert behandelt wird).

### 3.2.16 Beispiel

Horizontales scrollen aktivieren:

```
$('#example').dataTable( {
  "scrollX": true
} );
```

### 3.2.17 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- scrollY
- scrollCollapse

table.feature.scrollY

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Aktiviert vertikales scrollen. Vertikales scrollen beschränkt den DataTable auf die gegebene Höhe und aktiviert scrollen für alle Daten, die über den momentanen Viewport hinausgehen. Dies kann als Alternative zu Seiten verwendet werden, wenn eine große Menge an Daten in einem kleinen Bereich dargestellt werden soll (obwohl beide Möglichkeiten gleichzeitig aktiviert werden können, wenn dies gewollt ist).

Der angegebene Wert kann eine beliebige CSS-Einheit sein, oder eine Zahl (welche als Pixelwert interpretiert wird), und wird auf den Tabellenkörper angewandt (die Höhe von Tabellenkopf und -fuß werden also nicht direkt beachtet).

### 3.2.18 Beispiel

Aktiviere vertikales scrollen ohne Paginierung:

```
$('#example').dataTable( {
  "scrollY": "200px",
  "scrollCollapse": true,
  "paging": false
} );
```

### 3.2.19 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- paging
- scrollX
- scrollCollapse

table.feature.searching

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Diese Option erlaubt das aktivieren und deaktivieren der Suchfunktion von DataTables. Die Suche in DataTables ist "intelligent", insofern, dass sie die Eingabe von mehreren Worten (durch Leertasten getrennt) durch den Endnutzer erlaubt, und alle Reihen findet, die diese Worte enthalten, selbst wenn sie in einer anderen Reihenfolge sind (Dies erlaubt die Suche über mehrere Spalten hinweg).

Es sei bitte zu beachten, dass die Suche in DataTables eigentlich ein Filter ist, da sie substraktiv ist und Daten aus dem Datenset entfernt, wenn die Eingabe komplexer wird. Trotzdem wird sie hier und in der restlichen DataTables API als "Suche" bezeichnet, um konsistent zu bleiben und Konflikte mit Methoden mit ähnlichen Namen zu vermeiden (insbesondere der filter() API-Methode).

Es ist anzumerken, dass diese Option true bleiben muss, wenn die Suchfunktionen von DataTables verwendet werden sollen. Um das Eingabefeld für die Suche zu entfernen ohne die Suchfähigkeiten zu verlieren (z.B. um die search() API-Methode benutzen zu können), sollte die dom Option verwendet werden.

### 3.2.20 Beispiel

Suchfunktion im DataTable deaktivieren:

```
$('#example').dataTable( {
  "searching": false
} );
```

### 3.2.21 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- column().search()

Options

- columns.searchable
- search
- searchCols

table.feature.serverSide

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

DataTables hat zwei fundamentale Ausführungsmodi:

- Klientseitige Bearbeitung - wo Berechnungen für Filterung, Seitenbildung und Sortierung alle im Web-Browser durchgeführt werden.
- Serverseitige Bearbeitung - wo Berechnungen für Filterung, Seitenbildung und Sortierung alle von einem Server durchgeführt werden.

DataTables wird standardmäßig im Modus der klientseitigen Bearbeitung ausgeführt, kann mit dieser Option aber auf serverseitige Bearbeitung umgeschaltet werden. Serverseitige Bearbeitung ist nützlich, wenn mit großen Datensets (normalerweise >50.000 Einträge) gearbeitet wird. So kann eine Datenbank-Engine die Berechnungen für Sortierung etc. durchführen, wofür moderne Datenbank-Engines hochgradig optimiert sind. Dies ermöglicht die Verwendung von DataTables mit massiven Datensets (Millionen an Zeilen).

Bei der Ausführung im Modus serverseitiger Bearbeitung schickt DataTables Parameter an den Server, welche die benötigten Daten spezifizieren (welche Seite, welche angewandten Filter etc.), und erwartet bestimmte Parameter

als Rückgabe, damit die Tabelle korrekt dargestellt werden kann. Das von DataTables verwendete Klient-Server Kommunikationsprotokoll wird in der DataTables Dokumentation beschrieben.

### 3.2.22 Beispiel

Aktiviere serverseitige Bearbeitung:

```
$('#example').dataTable( {
  "serverSide": true,
  "ajax": "xhr.php"
} );
```

### 3.2.23 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax

table.feature.stateSave

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Aktiviert oder deaktiviert Zustandsspeicherung. Ist sie aktiviert, speichert DataTables Informationen wie Seitenposition, Anzeigelänge, Filtrierung und Sortierung. Aktualisiert der Endnutzer die Seite, so wird der Zustand der Tabelle auf den vorherigen Status gesetzt.

Für die Speicherung der Zustandsinformationen im Browser wird entweder die localStorage oder die sessionStorage HTML5 API verwendet. Die stateDuration Option sagt DataTables, welche API verwendet werden soll (localStorage: 0 oder größer, sessionStorage: -1).

Um die Zustandsdaten jeder Tabelle eindeutig zuordnen zu können, werden die Informationen mit einer Kombination aus der DOM id der Tabelle und dem Pfad der aktuellen Seite gespeichert. Ändert sich die id der Tabelle oder die URL der Seite, gehen die Informationen verloren.

Man beachte bitte, dass die Verwendung der HTML5 APIs für die Speicherung bedeutet, dass die eingebauten Optionen zum Speichern des Zustands **in IE6/7 nicht funktionieren**, da diese Browser diese APIs nicht unterstützen. Alternative Möglichkeiten, wie die Verwendung von Cookies oder dem Speichern des Zustands auf dem Server über Ajax, können durch die stateSaveCallback und stateLoadCallback Optionen verwendet werden.

### 3.2.24 Beispiel

Speichern des Zustands aktivieren:

```
$('#example').dataTable( {
  stateSave: true
} );
```

Speichern des Zustands aktivieren und Überschreiben der save/load-Handlers um nur die DOM id der Tabelle zu benutzen:

```
$('#example').dataTable( {
  stateSave: true,
  stateSaveCallback: function(settings,data) {
    localStorage.setItem( 'DataTables_' + settings.sInstance,
    JSON.stringify(data) )
  },
  stateLoadCallback: function(settings) {
    return JSON.parse( localStorage.getItem( 'DataTables_' + settings.sInstance ) )
  }
} );
```

### 3.2.25 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- state()
- state.clear()
- state.loaded()
- state.save()

#### Events

- stateLoaded
- stateLoadParams
- stateSaveParams

#### Options

- stateSaveCallback
- stateDuration
- stateLoadCallback
- stateLoadParams
- stateLoaded
- stateSaveParams

#### table.data

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

DataTables kann die im Tabellenkörper anzuzeigenden Daten von verschiedenen Quellen erhalten, eine davon ist ein durch diesen Initialisierungsparameter gegebenes Array an Zeilendaten. Wie bei anderen dynamischen Datenquellen können Arrays oder Objekte als Datenquellen der einzelnen Zeilen verwendet werden, wobei `columns.data` verwendet wird, um aus bestimmten Objekteigenschaften zu lesen.

Diese Initialisierungsoption kann nützlich sein, wenn eine Tabelle aus einer Javascript-Datenquelle oder einem selbsterstellten Ajax data get erstellt werden soll.

Es ist anzumerken, dass, wenn `data` spezifiziert ist, die Daten im Array jegliche Informationen im DOM der Tabelle bei der Initialisierung ersetzen werden.

### 3.2.26 Beispiel

Ein zweidimensionales Array als Datenquelle verwenden:

```
$('#example').dataTable( {
  "data": [
    [ "Tiger Nixon", "System Architect", "$3,120", "2011/04/25", "Edinburgh",
5421 ],
    [ "Garrett Winters", "Director", "$8,422", "2011/07/25", "Edinburgh", 8422 ],
    // ...
  ]
} );
```

Ein Array aus Objekten als Datenquelle verwenden:



```

$('#example').dataTable( {
  "data": [
    {
      "name":      "Tiger Nixon",
      "position":  "System Architect",
      "salary":    "$3,120",
      "start_date": "2011/04/25",
      "office":    "Edinburgh",
      "extn":      5421
    },
    {
      "name": "Garrett Winters",
      "position": "Director",
      "salary": "5300",
      "start_date": "2011/07/25",
      "office": "Edinburgh",
      "extn": "8422"
    },
    // ...
  ],
  "columns": [
    { "data": "name" },
    { "data": "position" },
    { "data": "office" },
    { "data": "extn" },
    { "data": "start_date" },
    { "data": "salary" }
  ]
} );

```

### 3.2.27 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- row.add()
- rows.add()

#### Options

- ajax
- columns.data

#### table.callback.createdRow

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dieser Callback wird ausgeführt wenn ein TR-Element erstellt wird (und alle TD-Elemente eingefügt wurden), oder, bei der Verwendung einer DOM-Quelle, wenn ein TR-Element registriert wird, und ermöglicht die Manipulation des TR-Elements.

Dies ist besonders nützlich, wenn deferred Rendering (deferRender ist true) oder serverseitige Bearbeitung (serverSide ist true) verwendet wird, um Events oder Informationen zum Klassennamen hinzuzufügen oder die Zeilen anderweitig bei der Erstellung zu formatieren.

### 3.2.28 Typ

Funktion `createdRow( row, data, dataIndex, cells )`

Parameter:

1. Name: row, Typ: node, Optional: Nein  
TR Zeilenelement welches soeben erstellt wurde.
2. Name: data, Typ: Array|Objekt, Optional: Nein  
Rohe Datenquelle (Array oder Objekt) dieser Zeile
3. Name: dataIndex, Typ: integer, Optional: Nein  
Der Index der Zeile im internen Speicher von DataTables
4. Name: cells, Typ: node[], Optional: Nein  
Seit 1.10.17: Die Zellen der Spalte.

### 3.2.29 Beispiel

Füge abhängig von den Daten der Zeile eine Klasse zur Zeile hinzu:

```
$('#example').dataTable( {
  "createdRow": function( row, data, dataIndex ) {
    if ( data[4] == "A" ) {
      $(row).addClass( 'important' );
    }
  }
} );
```

### 3.2.30 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- data
- deferRender
- serverSide
- columns.createdCell

## table.callback.drawCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Es kann nützlich sein, eine Aktion für jedes draw-Event der Tabelle ausführen zu können - zum Beispiel um einen externen Controller auf die neu angezeigten Daten zu aktualisieren, oder um bei serverseitiger Bearbeitung den neu erstellten Elementen Events zuzuweisen. Dieser Callback ist für genau solche Zwecke konzipiert und wird bei jedem draw ausgeführt.

### 3.2.31 Typ

Funktion drawCallback( settings )

Parameter:

Name: settings, Typ: DataTables.settings, Optional: Nein

DataTables Einstellungsobjekt

### 3.2.32 Beispiele

Melden, wenn DataTables einen draw durchführt:

```
$('#example').dataTable( {
  "drawCallback": function( settings ) {
    alert( 'DataTables has redrawn the table' );
  }
} );
```

API-Methoden im Callback benutzen um die Daten der Zeilen im draw zu erhalten:

```
$('#example').dataTable( {
  "drawCallback": function( settings ) {
    var api = this.api();

    // Output the data for the visible rows to the browser's console
    console.log( api.rows( {page:'current'} ).data() );
  }
} );
```

### 3.2.33 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- draw()

Events

- draw

table.callback.footerCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Identisch zu headerCallback, aber für die Fußzeile der Tabelle. Ermöglicht die modifizierung der Fußzeile bei jedem draw-Event.

Man beachte dass der Callback nie ausgeführt wird, wenn die Tabelle kein tfoot-Element hat.

### 3.2.34 Typ

Funktion footerCallback( tfoot, data, start, end, display )

Parameter:

1. Name: tfoot, Typ: node, Optional: Nein  
tfoot-Element der Fußzeile der Tabelle
2. Name: data, Typ: array, Optional: Nein  
Komplettes Datenarray der Tabelle. Dies ist in der Reihenfolge der Datenindices. Für die dargestellte Reihenfolge muss der display-Parameter benutzt werden
3. Name: start, Typ: integer, Optional: Nein  
Index des momentanen Startpunkts im display-Array.
4. Name: end, Typ: integer, Optional: Nein  
Index des momentanen Endpunkts im display-Array.
5. Name: display, Typ: array, Optional: Nein  
Index-Array um die visuelle Position auf das komplette Datenarray zu übersetzen.

### 3.2.35 Beispiel

Den Inhalt der Fußzeile im Callback verändern:

```
$('#example').dataTable( {
  "footerCallback": function( tfoot, data, start, end, display ) {
    $(tfoot).find('th').eq(0).html( "Starting index is "+start );
  }
} );
```

Die API benutzen um eine bestimmte Spalte aufzusummieren und das Ergebnis ausgeben:

```
$('#example').dataTable( {
  "footerCallback": function( tfoot, data, start, end, display ) {
    var api = this.api();
    $( api.column( 5 ).footer() ).html(
      api.column( 5 ).data().reduce( function ( a, b ) {
        return a + b;
      }, 0 )
    );
  }
} );
```

### 3.2.36 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `columns().footer()`
- `column().footer()`

Options

- `headerCallback`
- `drawCallback`

`table.callback.formatNumber`

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Separiert Tausenderstellen mit einem Komma ]

DataTables zeigt an verschiedenen Stellen Zahlen an, wenn Informationen über eine Tabelle gezeichnet werden. Beispiele sind das Informationselement der Tabelle und die Seitenkontrollen. Bei der Arbeit mit großen Zahlen ist es oft nützlich, die Tausenderstellen zwecks Lesbarkeit zu trennen- so wird z.B. 1 Millionen als "1,000,000" gerendert, wodurch der Nutzer schnell die Größenordnung der Zahl erkennen kann.

Diese Funktion erlaubt komplette Kontrolle über die Durchführung dieser Formatierung. Standardmäßig verwendet DataTables das Zeichen, das in `language.thousands` definiert wurde (welches standardmäßig ein Komma ist), um die Tausenderstellen zu trennen.

### 3.2.37 Type

#### Funktion `formatNumber( toFormat )`

Parameter:

Name: `toFormat`, Typ: `integer`, Optional: Nein

Zu formatierende Zahl.

Rückgabe:

`string`

Formatierter String, mit dem DataTables die Zahl anzeigt.

### 3.2.38 Beispiel

Stelle große Zahlen mit einem '-'Separator dar:

```
$('#example').dataTable( {
  "formatNumber": function ( toFormat ) {
    return toFormat.toString().replace(
      /B(?=(d{3})+(?!d))/g, "-"
    );
  };
} );
```

### 3.2.39 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- `language.info`
- `language.thousands`
- `infoCallback`

#### `table.callback.headerCallback`

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Funktion wird bei jedem draw-Event ausgelöst (also wenn ein Filter- Sortierungs- oder Seitenevent vom Endnutzer oder der API eingeleitet wird) und ermöglicht das dynamische Modifizieren der Kopfzeile. So können nützliche Informationen über die Tabelle berechnet und dargestellt werden.

### 3.2.40 Typ

Funktion `headerCallback( thead, data, start, end, display )`

Parameter:

1. Name: `thead`, Typ: `node`, Optional: Nein  
THEAD-Element der Kopfzeile der Tabelle
2. Name: `data`, Typ: `array`, Optional: Nein  
Komplettes Datenarray der Tabelle. Dies ist in der Reihenfolge der Datenindices. Für die dargestellte Reihenfolge muss der `display`-Parameter benutzt werden
3. Name: `start`, Typ: `integer`, Optional: Nein  
Index des momentanen Startpunkts im `display`-Array.
4. Name: `end`, Typ: `integer`, Optional: Nein  
Index des momentanen Endpunkts im `display`-Array.
5. Name: `display`, Typ: `array`, Optional: Nein  
Index-Array um die visuelle Position auf das komplette Datenarray zu übersetzen.

### 3.2.41 Beispiel

Zeige Informationen über die angezeigten Einträge:

```
$('#example').dataTable( {
  "headerCallback": function( thead, data, start, end, display ) {
    $(thead).find('th').eq(0).html( 'Displaying '+(end-start)+' records' );
  }
} );
```

### 3.2.42 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `columns().header()`
- `column().header()`

Options

- footerCallback
- drawCallback

### table.callback.infoCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Informationselement kann verwendet werden, um Informationen über den momentanen Status der Tabelle auszugeben. Obwohl die von DataTables zur Verfügung gestellten Optionen zur Internationalisierung mit den meisten Anpassungen umgehen können, soll der String möglicherweise noch weiter angepasst werden. In solchen Fällen kann dieser Callback verwendet werden.

Es ist anzumerken, dass diese Callback-Funktion nicht auslöst, wenn die info Option bei der Initialisierung deaktiviert wurde.

### 3.2.43 Typ

Funktion infoCallback( settings, start, end, max, total, pre )

Parameter:

1. Name: settings, Typ: DataTables.Settings, Optional: Nein  
DataTables Einstellungsobjekt
2. Name: start, Typ: integer, Optional: Nein  
Startposition in den Daten für den draw
3. Name: end, Typ: integer, Optional: Nein  
Endposition in den Daten für den draw
4. Name: max, Typ: integer, Optional: Nein  
Gesamtzahl der Zeilen in der Tabelle (ungeachtet der Filterung)
5. Name: total, Typ: integer, Optional: Nein  
Gesamtzahl der Zeilen im Datenset, nach der Filterung
6. Name: pre, Typ: string, Optional: Nein  
Der String den DataTables mit seinen eigenen Regeln formatiert hat

Rückgabe:

string

Der im Informationselement darzustellende String.

### 3.2.44 Beispiel

Zeige Start- und Endindex im Informationselement:



```
$('#example').dataTable( {
  "infoCallback": function( settings, start, end, max, total, pre ) {
    return start + " to " + end;
  }
});
```

Nutze die API im Callback um die Seitenzahl anzuzeigen:

```
$('#example').dataTable( {
  "infoCallback": function( settings, start, end, max, total, pre ) {
    var api = this.api();
    var pageInfo = api.page.info();

    return 'Page ' + (pageInfo.page+1) + ' of ' + pageInfo.pages;
  }
});
```

### 3.2.45 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- page.info()

Options

- info
- language.info

#### table.callback.initComplete

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagnworte / Tags</b>	

Es kann nützlich sein zu wissen, wann eine Tabelle fertig initialisiert, mit Daten gefüllt und gezeichnet wurde, besonders wenn eine Ajax-Datenquelle verwendet wird. In einem solchen Fall schließt die Tabelle ihren initialen Lauf ab, bevor die Daten geladen wurden (Ajax ist schließlich asynchron!). Also wird dieser Callback zur Verfügung gestellt, um anzuzeigen, dass die Daten fertig geladen wurden.

Weiterhin werden dem Callback die JSON-Daten übergeben, die beim Laden von Ajax-Daten vom Server empfangen wurden, was bei der Konfiguration von Komponenten, die in Verbindung mit der Tabelle stehen, nützlich sein kann.

### 3.2.46 Typ

#### Funktion `initComplete( settings, json )`

Parameter:

1. Name: `settings`, Typ: `DataTables.Settings`, Optional: Nein  
DataTables Einstellungsobjekt
2. Name: `json`, Typ: `objekt`, Optional: Nein  
JSON-Daten die bei gesetzter Ajax Option vom Server empfangen wurden. Sonst undefiniert.

### 3.2.47 Beispiel

Eine Benachrichtigung anzeigen wenn die Tabelle fertig geladen wurde:

```
$('#example').dataTable( {
  "initComplete": function(settings, json) {
    alert( 'DataTables has finished its initialisation.' );
  }
} );
```

Eine Ladenachricht anzeigen während die Tabelle geladen wird (processing kann dies auch):

```
$('#<div class="loading">Loading</div>').appendTo('body');

$('#example').dataTable( {
  "initComplete": function( settings, json ) {
    $('#div.loading').remove();
  }
} );
```

### 3.2.48 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Events

- `init`

Options

- `ajax`
- `drawCallback`

`table.callback.preDrawCallback`

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Gegenstück zur `drawCallback`-Funktion, diese Funktion wird am Anfang von jedem `draw` einer Tabelle aufgerufen. Sie kann also benutzt werden um die Anzeige vor dem `draw` zu aktualisieren oder zu säubern (zum Beispiel durch das Entfernen von Events), und kann außerdem den `draw` abbrechen, indem sie `false` zurückgibt. Bei jedem anderen Rückgabewert (auch `undefined`) findet der komplette `draw` statt.

### 3.2.49 Typ

Funktion `preDrawCallback( settings )`

Parameter:

Name: `settings`, Typ: `DataTables.Settings`, Optional: Nein

`DataTables` Einstellungsobjekt

### 3.2.50 Beispiel

Entferne gebundene Events von den Zellen im Tabellenkörper:

```
$('#example').dataTable( {
  "preDrawCallback": function( settings ) {
    $('#example tbody').off( 'click', 'td' );
  }
} );
```

Breche den `draw` der Tabelle ab wenn `#test` den Wert 1 hat:

```
$('#example').dataTable( {
  "preDrawCallback": function( settings ) {
    if ( $('#test').val() == 1 ) {
      return false;
    }
  }
} );
```

### 3.2.51 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- `drawCallback`

## table.callback.rowCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dieser Callback ermöglicht 'nachbearbeitung' jeder Zeile nachdem sie vom jeweiligen draw der Tabelle generiert wurde, aber *bevor* sie ins Dokument gerendert wird. Das bedeutet, dass der Inhalt der Zeile möglicherweise noch keine Dimensionen hat (zum Beispiel `$.width()`), wenn die Zeile nicht schon im Dokument vorhanden ist.

Diese Funktion kann benutzt werden, um den Klassennamen der Zeile zu setzen oder das tr-Element der Zeile anderweitig zu manipulieren (wobei anzumerken ist, dass `createdRow` oftmals effizienter sein kann).

### 3.2.52 Typ

Funktion `rowCallback( row, data, displayNum, displayIndex, dataIndex )`

Parameter:

1. Name: `row`, Typ: `node`, Optional: Nein  
TR-Element, welches in das Dokument eingefügt wird.
2. Name: `data`, Typ: `array,objekt`, Optional: Nein  
Datenquelle der Zeile. **Wichtig:** Dieser Parameter ist die originale Datenquelle, die für diese Zeile benutzt wird. Werden Objekte verwendet, ist `data` ein Objekt - werden Arrays verwendet, ist `data` ein Array. Wie die Daten aus diesem Parameter beschaffen werden hängt also von der Konfiguration der Tabelle ab.
3. Name: `displayNum`, Typ: `integer`, Optional: Nein  
Zeilennummer auf der aktuellen Seite an dargestellten Zeilen
4. Name: `displayIndex`, Typ: `integer`, Optional: Nein  
Zeilennummer in der momentanen Suchmenge (also über alle verfügbaren Seiten hinweg).
5. Name: `datalIndex`, Typ: `integer`, Optional: Nein  
Interner Index der Zeile von `DataTables` - siehe `row().index()`

### 3.2.53 Beispiele

Zellen anhand ihres Inhalts hervorheben (Objekt-Datenquelle):

```

$('#example').dataTable( {
  "rowCallback": function( row, data ) {
    if ( data.grade == "A" ) {
      $('td:eq(4)', row).html( '<b>A</b>' );
    }
  }
} );

```

Zellen anhand ihres Inhalts hervorheben (Array-Datenquelle):

```
$('#example').dataTable( {
  "rowCallback": function( row, data ) {
    if ( data[4] == "A" ) {
      $('td:eq(4)', row).html( '<b>A</b>' );
    }
  }
} );
```

### 3.2.54 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

drawCallback  
createdRow

table.callback.stateLoadCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mit diesem Callback kann definiert werden von wo - und wie - der Zustand der Tabelle geladen wird. Standardmäßig lädt DataTables aus der localStorage oder sessionStorage, aber für permanenteren Speicher kann der Zustand auch in einer serverseitigen Datenbank gespeichert werden.

Vor DataTables 1.10.13 musste diese Funktion synchron aufgerufen werden, der Zustand wurde also von der Funktion zurückgegeben. Seit 1.10.13 ist es möglich, den Zustand asynchron mit Ajax oder einer beliebigen anderen asynchronen Methode zu laden und die Callback-Funktion auszuführen, wobei der geladene Zustand übergeben wird.

Aus Kompatibilitätsgründen kann der Zustand weiterhin synchron zurückgegeben werden. Um die Callback-Funktion zu benutzen, sollte die stateLoadCallback-Funktion keinen Rückgabewert haben. Siehe unten für Beispiele beider Fälle.

Es ist anzumerken, dass dieser Callback Hand in Hand mit stateSaveCallback arbeitet. Dieser Callback lädt den Zustand aus dem Speicher, wenn die Tabelle neu geladen wird, während stateSaveCallback ihn speichert.

### 3.2.55 Typ

#### Funktion stateLoadCallback( settings, callback )

Parameter:

1. Name: settings, Typ: DataTables.Settings, Optional: Nein  
DataTables Einstellungsobjekt
2. Name: callback, Typ: Funktion, Optional: Nein  
**Seit 1.10.13:** Callback-Funktion die ausgeführt werden soll wenn die Zustandsdaten bereit sind nachdem sie mit Ajax oder einer anderen asynchronen Methode geladen wurden. Soll diese Option verwendet werden, so muss stateLoadCallback undefined als Rückgabewert haben (also einfach nichts zurückgeben)!

Returns:

objekt, undefined

Wenn die Daten synchron geladen werden, sollte der Rückgabewert der geladene Zustand sein (oder null, wenn keine Daten geladen wurden).

Wenn die Daten asynchron geladen werden (z.B. mit Ajax), sollte undefined zurückgegeben werden (einfach kein return-Statement benutzen) und die Callback-Funktion aufgerufen werden, wenn der Zustand geladen wurde. Man beachte, dass diese Option DataTables 1.10.13 oder neuer benötigt.

### 3.2.56 Beispiel

Zustand von einem Server über Ajax laden (1.10.13 oder neuer):

```
$('#example').DataTable( {  
  stateSave: true,  
  stateLoadCallback: function (settings, callback) {  
    $.ajax( {  
      url: '/state_load',  
      dataType: 'json',  
      success: function (json) {  
        callback( json );  
      }  
    } );  
  }  
} );
```

Zustand von einem Server über Sjax laden (vor 1.10.13):

```

$('#example').DataTable( {
  stateSave: true,
  stateLoadCallback: function (settings) {
    var o;

    // Schicke ein Ajax-Request an den Server um die Daten zu kriegen. Es ist
    // anzumerken, dass
    // dies eine synchrone Anfrage ist, da die Daten den Rückgabewert
    // der Funktion bilden.
    $.ajax( {
      url: '/state_load',
      async: false,
      dataType: 'json',
      success: function (json) {
        o = json;
      }
    } );

    return o;
  }
} );

```

### 3.2.57 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- state()
- state.clear()
- state.loaded()
- state.save()

#### Events

- stateLoaded
- stateLoadParams
- stateSaveParams

#### Options

- stateSave
- stateSaveCallback
- stateLoadParams
- stateLoaded
- stateSaveParams

#### table.callback.stateLoaded

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Callback der feuert, sobald der Zustand geladen wurde (stateLoadCallback) und die gespeicherten Daten manipuliert wurden (falls nötig - stateLoadParams).

Dieser Callback ist nützlich wenn Informationen aus dem gespeicherten Zustand geholt werden sollen, ohne zu betrachten wo und wie der Zustand gespeichert wurde. Er kann zum Beispiel benutzt werden um eigene Filterinputs zu besiedeln.

### 3.2.58 Typ

Funktion stateLoaded( settings, data )

Parameter:

1. Name: settings, Typ: DataTables.Settings, Optional: Nein  
DataTables Einstellungsobjekt.
2. Name: data, Typ: Objekt, Optional: Nein  
Zu ladende Daten. Die Daten kommen von stateSaveParams.

### 3.2.59 Beispiel

Den gespeicherten globalen Filter anzeigen:

```

$('#example').dataTable( {
  "stateSave": true,
  "stateLoaded": function (settings, data) {
    alert( 'Saved filter was: '+data.search.search );
  }
} );

```

### 3.2.60 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- state()
- state.clear()
- state.loaded()
- state.save()

Events

- stateLoadParams
- stateSaveParams

Options



- stateSave
- stateSaveCallback
- stateLoadParams
- stateSaveParams
- stateLoadCallback

### table.callback.stateLoadParams

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Callback welcher die Modifizierung des gespeicherten Zustands vor dem Laden dieses Zustands ermöglicht. Dieser Callback wird aufgerufen wenn die Tabelle den Zustand aus den gespeicherten Daten lädt, aber bevor das Einstellungsobjekt vom gespeicherten Zustand modifiziert wird.

Es ist anzumerken, dass die stateLoadCallback Option verwendet wird, um zu definieren ob und wo der Zustand geladen wird, während diese Funktion die Daten manipuliert nachdem sie aus dem Speicher geladen wurden.

Weiter ist anzumerken, dass Plugin-Autoren das stateLoadParams-Event benutzen sollten um Parameter für ihr Plugin zu laden.

### 3.2.61 Typ

Funktion stateLoadParams( settings, data )

Parameter:

1. Name: settings, Typ: DataTables.Settings, Optional: Nein  
DataTables Einstellungsobjekt.
2. Name: data, Typ: Objekt, Optional: Nein  
Zu ladende Daten. Die Daten kommen von stateSaveParams.

### 3.2.62 Beispiel

Einen gespeicherten Filter entfernen, wodurch gespeicherte Filter nie geladen werden:

```
$('#example').dataTable( {
  "stateSave": true,
  "stateLoadParams": function (settings, data) {
    data.search.search = "";
  }
} );
```

Deaktivieren des Zustandsladens durch Zurückgeben von false:

```
$('#example').dataTable( {
  "stateSave": true,
  "stateLoadParams": function (settings, data) {
    return false;
  }
});
```

### 3.2.63 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- state()
- state.clear()
- state.loaded()
- state.save()

#### Events

- stateLoaded
- stateSaveParams

#### Options

- stateSave
- stateSaveCallback
- stateLoaded
- stateLoadCallback
- stateSaveParams

#### table.callback.stateSaveCallback

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

DataTables kann den Zustand der Tabelle speichern (Seite, Filterung, etc.) wenn die stateSave Option aktiviert ist, und verwendet dafür standardmäßig den localStorage von HTML5. Diese Callback-Methode ermöglicht das Ändern des Speicherorts (zum Beispiel zu einer serverseitigen Datenbank oder einem Cookie).

Das data, das der Funktion übergeben wird, ist ein Objekt mit der folgenden Struktur:

```

{
  "time": {number} // Zeitstempel des Erstellungszeitpunkts des
Objekts
  "start": {number} // Startpunkt der Anzeige
  "length": {number} // Seitenlänge
  "order": {array} // 2D-Array von Informationen über
Spaltenreihenfolge (siehe 'order' Option)
  "search": {
    "search": {string} // Suchbegriff
    "regex": {boolean} // Zeigt ob Suchbegriff als regex behandelt
werden soll oder nicht
    "smart": {boolean} // Flagge um intelligente Suche von DataTables
zu aktivieren
    "caseInsensitive": {boolean} // Flagge für Beachtung von Groß- und
Kleinschreibung
  },
  "columns" [
    {
      "visible": {boolean} // Sichtbarkeit der Spalte
      "search": {} // Objekt mit Suchinformationen der Spalte.
Gleiche Struktur wie 'search' oben.
    }
  ]
}

```

Es ist anzumerken, dass Erweiterungen zusätzliche Informationen zu dieser Struktur hinzufügen können, und auch über die `stateSaveParams` Option oder das `stateSaveParams` Event Parameter hinzugefügt werden können. Weiterhin sind die gespeicherten Informationen Typsensitiv. Der Datentyp der von DataTables gegebenen Daten muss also erhalten bleiben. Zum Beispiel muss der `start`-Parameter einen Wert vom Typ `number` haben.

Diese Methode muss nur die ihr übergebenen Daten speichern. Um die Daten zu manipulieren wird die `stateSaveParams` Methode verwendet.

Dieser Callback arbeitet Hand in Hand mit `stateLoadCallback`. Diese Methode speichert den Zustand, während `stateLoadCallback` ihn vom Speicherort dieses Callbacks lädt.

### 3.2.64 Typ

#### Funktion `stateSaveCallback( settings, data )`

Parameter:

1. Name: `settings`, Typ: `DataTable.Settings`, Optional: Nein  
DataTable Einstellungsobjekt.
2. Name: `data`, Typ: Objekt, Optional: Nein  
Zu speichernde Daten. Die Daten kommen von `stateSaveParams`.

### 3.2.65 Beispiel

Den Zustand über Ajax auf einem Server speichern:

```

$('#example').dataTable( {
  "stateSave": true,
  "stateSaveCallback": function (settings, data) {
    // Ein Ajax-Request mit dem Zustandsobjekt an den Server schicken
    $.ajax( {
      "url": "/state_save",
      "data": data,
      "dataType": "json",
      "type": "POST",
      "success": function () {}
    } );
  }
} );

```

### 3.2.66 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- state()
- state.clear()
- state.loaded()
- state.save()

#### Events

- stateLoaded
- stateLoadParams
- stateSaveParams

#### Options

- stateSave
- stateLoadParams
- stateLoaded
- stateLoadCallback
- stateSaveParams

#### table.deferLoading

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Bei der Verwendung von serverseitiger Bearbeitung ist die standardmäßige Vorgehensweise von DataTables, einfach alle Daten in der Tabelle zu verwerfen, und eine Anfrage an den Server zu schicken, um die erste anzuzeigende Seite zu erhalten. Dies ist für leere Tabellen in Ordnung, wird aber bereits die erste Seite der Daten in reinem HTML angezeigt, so ist es eine Verschwendung an Ressourcen. Daher existiert diese Option um DataTables anzuweisen, keine initiale Anfrage abzuschicken, sondern die bereits auf der Seite vorhandenen Daten zu verwenden (keine Sortierung etc. wird angewendet).

`deferLoading` wird verwendet um zu Zeigen, dass aufgeschobenes Laden benötigt wird, es sagt DataTables aber auch, wie viele Einträge die gesamte Tabelle hat (wodurch die Informations- und Seitenelemente korrekt angezeigt werden können). Für den Fall, dass beim initialen Laden ein Filter auf die Tabelle angewendet wird, kann dies mit einem Array als Parameter dargestellt werden. Dabei ist das erste Element die Anzahl der Einträge nach der Filterung und das zweite Element die Anzahl der Einträge ohne Filterung (wodurch das Informationselement korrekt angezeigt werden kann).

Es ist anzumerken, dass diese Option nur Auswirkungen hat wenn `serverSide` aktiviert ist. Bei klientseitiger Bearbeitung hat sie keinen Effekt.

### 3.2.67 Typen

#### integer

Als Integer gegeben aktiviert der Parameter verschobenes Laden und informiert DataTables über die Anzahl der Einträge im vollen Datenset.

#### array

Als Array gegeben aktiviert der Parameter auch verschobenes Laden, der erste Wert informiert DataTables über die Anzahl Einträge im gefilterten Ergebnis, und der zweite Eintrag über die Anzahl Einträge im vollen Datenset ohne Filterung.

### 3.2.68 Beispiel

57 Einträge verfügbar in der Tabelle, keine angewandten Filter:

```
$('#example').dataTable( {  
  "serverSide": true,  
  "ajax": "scripts/server_processing.php",  
  "deferLoading": 57  
} );
```

57 Einträge nach der Filterung, 100 ohne Filterung (ein initialer Filter angewandt):

```
$('#example').dataTable( {  
  "serverSide": true,  
  "ajax": "scripts/server_processing.php",  
  "deferLoading": [ 57, 100 ],  
  "search": {  
    "search": "my_filter"  
  }  
} );
```

### 3.2.69 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax
- serverSide

table.destroy

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Initialisiere einen neuen DataTable wie gewöhnlich, existiert aber bereits ein DataTable mit dem gegebenen Selektor, so wird dieser zerstört und durch die neue Tabelle ersetzt. Dies kann nützlich sein wenn eine Eigenschaft einer Tabelle verändert werden soll, die nicht durch die API verändert werden kann.

Es ist anzumerken, dass, wenn die Konfiguration der Tabelle nicht verändert wird, sondern nur die anzuzeigenden Daten, die Verwendung von `ajax.reload()` (oder `rows.add()` etc.) deutlich effizienter ist.

### 3.2.70 Beispiel

Zerstöre eine bestehende Tabelle und erstelle eine neue:

```
$('#example').dataTable( {
  "scrollY": "200px"
} );

// Irgendwann später, erstelle ohne scrollY neu
$('#example').dataTable( {
  "destroy": true
} );
```

### 3.2.71 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- ajax.reload()

## Options

- retrieve

## table.displayStart

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Definiert den Startpunkt für die Anzeige der Daten wenn DataTables mit paginierung verwendet wird (paging Option).

Es ist anzumerken, dass dieser Parameter die Nummer des ersten anzuzeigenden Eintrags ist (von 0 angefangen) und keine Seitennummer. Hat man also 10 Einträge pro Seite und will auf der dritten Seite anfangen, so sollte der Wert 20 sein, nicht 2 oder 3.

### 3.2.72 Beispiel

Zeige beim Start Zeile 20 als erste Zeile in der Tabelle:

```
$('#example').dataTable( {
  "displayStart": 20
} );
```

### 3.2.73 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

## API

- page()
- page.info()

## Options

- lengthChange
- paging

## table.dom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'lfrtip' ] (Wird von Möglichkeiten für Integration von Styling wie Bootstrap, Foundation oder jQuery UI verändert. Siehe Beschreibung für die gesetzten Werte.)

DataTables fügt eine Reihe an Elementen zur Kontrolle und Darstellung von Informationen zu Tabellen hinzu. Die Positionen dieser Elemente auf dem Bildschirm werden durch eine Kombination aus ihrer Reihenfolge im Dokument (DOM) und dem auf die Elemente angewandten CSS kontrolliert. Dieser Parameter wird verwendet, um ihre Reihenfolge und zusätzliches umschließendes Markup im DOM zu kontrollieren.

Jedes Kontrollelement der Tabelle hat in DataTables einen zugewiesenen Buchstaben. Dieser Buchstabe wird in der dom Option verwendet, um anzuzeigen, wo dieses Element in der Reihenfolge der Elemente des Dokuments steht.

## Optionen

- l - Eingabe zur Veränderung der Länge
- f - Eingabe zur Filterung
- t - Die Tabelle!
- i - Zusammenfassung der Tabelleninformationen
- p - Paginierungskontrolle
- r - Anzeigeelement der Verarbeitung

Jede der Optionen (außer der Tabelle selbst) kann mehrfach angegeben werden. So können Kontrollelemente über und unter der Tabelle angezeigt werden. DataTables synchronisiert mehrfach vorhandene Elemente automatisch.

Zusätzlich zu den oben genannten Optionen versteht DataTables die folgenden Konstanten (Es ist anzumerken, dass diese beiden Optionen in 1.10 deprecated sind und in 1.11 entfernt werden, wenn die jQueryUI Option in ihr eigenes Plug-in verschoben wird.):

- H - jQueryUI theme "header"-Klassen  
(fg-toolbar ui-widget-header ui-corner-tl ui-corner-tr ui-helper-clearfix)
- F - jQueryUI theme "footer"-Klassen  
(fg-toolbar ui-widget-header ui-corner-bl ui-corner-br ui-helper-clearfix)

## Markup

Weiterhin können auch zusätzliche div-Elemente in das Dokument eingefügt werden, welche für das Styling/ die Verschachtelung der Kontrollelemente verwendet werden können. Für div-Tags wird die folgende Syntax erwartet:

- < und > - div-Element
- <"class" und > - div mit einer Klasse
- <"#id" und > - div mit einer ID
- <"#id.class" und > - div mit einer ID *und* einer Klasse



## Styling

Die Styling-Bibliotheken, die DataTables unterstützt, überschreiben den standardmäßigen Wert der dom Option und ersetzen ihn durch einen mit ihrem Layout-System kompatiblen Wert. Die Bootstrap-Integration nutzt zum Beispiel Bootstraps Gitterlayout aus.

Die Standardwerte der verschiedenen Styling-Integrationen sind:

Bootstrap 3:

```
"<'row'<'col-sm-6'l><'col-sm-6'f>>" +
"<'row'<'col-sm-12'tr>>" +
"<'row'<'col-sm-5'i><'col-sm-7'p>>",
```

Bootstrap 4:

```
"<'row'<'col-sm-12 col-md-6'l><'col-sm-12 col-md-6'f>>" +
"<'row'<'col-sm-12'tr>>" +
"<'row'<'col-sm-12 col-md-5'i><'col-sm-12 col-md-7'p>>",
```

Foundation:

```
"<'row'<'small-6 columns'l><'small-6 columns'f>r>" +
"t"+
"<'row'<'small-6 columns'i><'small-6 columns'p>>",
```

jQuery UI:

```
'<"fg-toolbar ui-toolbar ui-widget-header ui-helper-clearfix ui-corner-tl ui-corner-
tr"lfr>' +
't'+
'<"fg-toolbar ui-toolbar ui-widget-header ui-helper-clearfix ui-corner-bl ui-corner-
br"ip>',
```

Semantic UI:

```
"<'ui stackable grid'" +
  "<'row'" +
    "<'eight wide column'l>" +
    "<'right aligned eight wide column'f>" +
  ">" +
  "<'row dt-table'" +
    "<'sixteen wide column'tr>" +
  ">" +
  "<'row'" +
    "<'seven wide column'i>" +
    "<'right aligned nine wide column'p>" +
  ">" +
  ">"
```

## Plug-ins

DataTables Feature Plug-ins können entwickelt werden, um zusätzliche Funktionalitäten zu DataTables hinzuzufügen, und verwenden dabei oft diese Option, um Buchstaben für die neuen Features in DataTables einzufügen. Die Buttons-Erweiterung fügt zum Beispiel die Option 'B' zu dom hinzu, um anzuzeigen, wo Knöpfe zur Kontrolle der Tabelle eingefügt werden sollen.

Die folgenden Erweiterungen können durch die dom Option initialisiert werden:

- B - Buttons
- S - Select

### 3.2.74 Beispiel

Keine filternde Eingabekontrolle:

```

/* Resultiert in:
  <div class="wrapper">
    {length}
    {processing}
    {table}
    {information}
    {pagination}
  </div>
*/
$('#example').dataTable( {
  "dom": 'lrtip'
} );

```

Einfacheres Wrapper-Element:

```

/* Resultiert in:
  <div class="wrapper">
    {filter}
    {length}
    {information}
    {pagination}
    {table}
  </div>
*/
$('#example').dataTable( {
  "dom": '<"wrapper"flipt>'
} );

```

Länge und Filter über, Information und Paginierung unter der Tabelle:

```

/* Resultiert in:
  <div>
    {length}
    {filter}
    <div>
      {table}
    </div>
    {information}
    {pagination}
  </div>
*/
$('#example').dataTable( {
  "dom": '<lf<t>ip>'
} );

```

Tabellenzusammenfassung in der Kopfzeile, Filtrierung, Länge und Seite in der Fußzeile, mit einem leerenden Element:

```

/* Resultiert in:
  <div class="top">
    {information}
  </div>
  {processing}
  {table}
  <div class="bottom">
    {filter}
    {length}
    {pagination}
  </div>
  <div class="clear"></div>
*/
$('#example').dataTable( {
  "dom": '<"top"i>rt<"bottom"flp><"clear">'
} );

```

### 3.2.75 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### Options

- searching
- info
- ordering
- processing
- paging
- lengthChange

## table.lengthMenu

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 25, 50, 100] ]

Dieser Parameter ermöglicht das Spezifizieren der Einträge in der der ausklappbaren select-Liste der Seitenlängen, die DataTables anzeigt, wenn Paginierung aktiviert ist. Möglich Werte sind:

- Eindimensionales Integer-Array, welches sowohl für die dargestellten Optionen als auch für die entsprechenden Werte der Seitenlänge benutzt wird.
- Zweidimensionales Array, welches das erste innere Array als Werte der Seitenlänge und das zweite innere Array als anzuzeigende Optionen verwendet. Dies ist nützlich für die Verwendung von Strings wie "Alle".

Die Werte der Seitenlänge müssen immer ganze Zahlen > 0 sein, mit der Ausnahme von -1. Wird -1 als Wert der Seitenlänge ausgewählt, so deaktiviert DataTables die Paginierung (zeigt also alle Zeilen an).

Es ist anzumerken, dass die `pageLength` Option, falls sie nicht spezifiziert ist, auf den ersten Wert des Arrays der Seitenlängen gesetzt wird.

### 3.2.76 Beispiel

Zeige Optionen für 10, 25, 50, 75 und 100 Einträge:

```
$('#example').dataTable( {
  "lengthMenu": [ 10, 25, 50, 75, 100 ]
} );
```

Zeige Optionen für 10, 25, 50 und alle Einträge:

```
$('#example').dataTable( {
  "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, "Alle"] ]
} );
```

### 3.2.77 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `page()`
- `page.info()`

- `page.len()`

#### Options

- `paging`
- `lengthChange`
- `pageLength`

#### table.order

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: `[[0, "asc"]]` ]

Wenn die Sortierung aktiviert ist (ordering Option), wird DataTables während der Initialisierung einen ersten Sortierlauf durchführen. Mit diesem Parameter kann definiert werden auf welchen Spalten die Sortierung durchgeführt wird, sowie die Sortierungsrichtung. Die order Option muss ein Array aus Arrays sein, in dem jedes innere Array zwei Elemente enthält:

- Spaltenindex auf dem Sortiert werden soll
- Zu verwendende Sortierungsrichtung ("`asc`" für aufsteigend oder "`desc`" für absteigend).

Diese zweidimensionale Array-Struktur ermöglicht die Definierung einer Sortierordnung über mehrere Spalten als initialer Zustand, falls erwünscht.

### 3.2.78 Beispiele

Keine Sortierung wird während der Initialisierung angewandt. Die Zeilen werden in der Reihenfolge angezeigt, in der sie von DataTables gelesen wurden (also der Reihenfolge aus dem DOM falls DOM als Quelle benutzt wird, oder dem Datenarray falls Ajax oder data als Quelle benutzt wird):

```
$('#example').dataTable( {
  "order": []
} );
```

Sortierung über mehrere Spalten als initialer Zustand:

```
$('#example').dataTable( {
  "order": [[ 0, 'asc' ], [ 1, 'asc' ]]
} );
```

### 3.2.79 Verwandt

#### API

- order()

#### Options

- ordering
- orderFixed
- columns.orderable
- columns.orderData
- columns.orderSequence

#### Erlaubte Werte

- {integer}
- asc
- desc

#### table.orderCellsTop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ermöglicht Kontrolle darüber, ob DataTables die obere (true) einzigartige in der Spalte gefundene Zelle oder die untere (false) benutzt, um den Standard-Sortierungslister einzubinden. Dies ist bei der Verwendung von komplexen Headern nützlich.

Man betrachte zum Beispiel den folgenden HTML-Header:

```
<thead>
  <tr>
    <td rowspan="2">1</td>
    <td>2.1</td>
  </tr>
  <tr>
    <td>2.2</td>
  </tr>
</thead>
```

In diesem Fall werden mit orderCellsTop auf false (Standardwert) die Zellen 1 und 2.2 die Sortierungslister eingebunden kriegen. Ist orderCellsTop hingegen auf true, so werden sie auf 1 und 2.1 eingebunden.

### 3.2.80 Beispiel

Verwende die oberen Zellen in einem komplexen Header für die Sortierung:

```
$('#example').dataTable( {
  "orderCellsTop": true
} );
```

### 3.2.81 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- order()

Options

- ordering
- order

table.orderClasses

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

DataTables hebt die Spalten hervor, die benutzt werden um den Inhalt des Tabellenkörpers zu sortieren. Den Zellen dieser Spalten werden Klassen hinzugefügt, auf die CSS angewandt wird, um die Zellen hervorzuheben.

Die Klassen die zu diesem Zweck zu den sortierenden Spalten hinzugefügt werden sind `sorting_1`, `sorting_2` und `sorting_3`. Die Zahl zeigt die Sortierungsstufe an, wenn über mehrere Spalten sortiert wird. Wenn über mehr als 3 Spalten sortiert wird, wird `sorting_3` mehrmals verwendet.

Es ist anzumerken, dass dieses Feature die Performanz beeinträchtigen kann, besonders in alten Browsern und wenn viele Zeilen dargestellt werden müssen, da dabei eine große Anzahl DOM-Objekte manipuliert wird. Daher kann das Feature über diese Option für alte Browser oder große Datensets deaktiviert werden.

### 3.2.82 Beispiel

Klassen für sortierende Spalten deaktivieren:

```
$('#example').dataTable( {
  "orderClasses": false
} );
```

### 3.2.83 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- order()

Options

- ordering
- order

table.orderFixed

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option funktioniert ähnlich zur order-Option, welche einen initialen Sortierungszustand für die Tabelle definiert, der dann vom Nutzer durch Klicken auf die Spaltenköpfe verändert werden kann. Die Sortierung dieser Option hingegen wird immer auf die Tabelle angewandt, ungeachtet der Nutzerinteraktion.

Diese fixe Sortierung kann angewendet werden bevor (pre) oder nachdem (post) die Tabelle durch den Nutzer sortiert wurde. Sie kann in zwei Formen (Array oder Objekt) angegeben werden.

Die Werte, die verwendet werden, um die Sortierbedingungen der Tabelle zu beschreiben, sind als Arrays mit zwei Elementen gegeben:

- Spaltenindex nach dem sortiert werden soll.
- Richtung nach der sortiert werden soll ('asc' für aufsteigend, 'desc' für absteigend)

Es ist auch möglich eine Menge an verschachtelten Arrays (also Arrays in Arrays) anzugeben, um eine Sortierung über mehrere Spalten zu definieren.

Diese Option kann nützlich sein wenn eine Spalte existiert (sichtbar oder unsichtbar), nach der immer als erstes sortiert werden muss - eine Prioritäten- oder Indexspalte zum Beispiel, oder um ähnliche Zeilen zu Gruppieren.



## 3.2.84 Typen

### Array

Präfix-Sortierung.

Als Array gegeben wird die durch orderFixed spezifizierte Sortierung als Präfix-Sortierung angewendet. Das Format des Arrays ist oben beschrieben.

### Objekt

Präfix- und / oder Postfix-Sortierung.

Als Objekt gegeben kann der pre- und / oder post-Parameter verwendet werden, um jeweils Präfix- und / oder Postfix-Sortierung zu spezifizieren. Keiner der beiden Parameter ist notwendig, wodurch nur das benötigte spezifiziert werden muss.

Wie bei der Typisierung als Array ist das Format der zu verwendenden Arrays oben beschrieben.

## 3.2.85 Beispiele

Es wird immer zuerst nach der ersten Spalte sortiert:

```
$('#example').dataTable( {  
  "orderFixed": [ 0, 'asc' ]  
} );
```

Wie oben, nur in Objektform:

```
$('#example').dataTable( {  
  "orderFixed": {  
    "pre": [ 0, 'asc' ]  
  }  
} );
```

Es wird immer als letztes nach der ersten und zweiten Spalte sortiert (postfix) - ein 2D-Array wird verwendet um mehrere Spalten für die Sortierung anzugeben:

```
$('#example').dataTable( {  
  "orderFixed": {  
    "post": [[ 0, 'asc' ], [ 1, 'asc' ]]  
  }  
} );
```

Sowohl Postfix- als auch Präfix-Optionen spezifiziert:

```
$('#example').dataTable( {
  "orderFixed": {
    "pre": [ 0, 'asc' ],
    "post": [ 1, 'asc' ]
  }
} );
```

### 3.2.86 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `order()`
- `order.fixed()`

#### Options

- `ordering`
- `order`
- `columns.orderable`
- `columns.orderData`

#### table.orderMulti

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Wenn sortierung (`ordering`) aktiviert ist, erlaubt DataTables es standardmäßig den Nutzern, durch einen Shift-Klick auf die einzelnen Spaltenköpfe Sortierungen über mehrere Spalten zu definieren. Dies kann zwar für Nutzer nützlich sein, erhöht aber auch die Komplexität und damit potentiell die Verarbeitungszeit der Sortierung. Daher wird diese Option bereitgestellt, mit der die Sortierung über mehrer Spalten durch den Nutzer deaktiviert werden kann.

Es ist anzumerken, dass das deaktivieren dieser Option nicht den Entwickler an der Definition von Sortierungen über mehrere Zeilen bei `columns.orderData`, `order`, oder der API-Methode `order()` hindert. Es verhindert nur, dass der Nutzer solche Sortierungen definiert.

### 3.2.87 Beispiel

Sortieren über mehrere Spalten für den Nutzer deaktivieren:

```
$('#example').dataTable( {
  "orderMulti": false
} );
```

### 3.2.88 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- order()
- column().order()
- columns().order()

#### Options

- columns.orderData
- columns.orderable
- ordering
- order
- orderFixed

#### table.pageLength

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Anzahl der auf einer Seite anzuzeigenden Zeilen wenn Paginierung aktiviert ist.

Ist lengthChange aktiviert (was es standardmäßig ist), so kann der Endnutzer den hier gesetzten Wert über ein Pop-up-Menü überschreiben (siehe lengthMenu).

### 3.2.89 Beispiel

Zeige 50 Einträge pro Seite an:

```
$('#example').dataTable( {
  "pageLength": 50
} );
```

### 3.2.90 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `page()`
- `page.info()`
- `page.len()`

#### Options

- `lengthChange`
- `lengthMenu`
- `paging`

#### table.pagingType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "full\_numbers" ]

Die Paginierungsoption von DataTables zeigt unter der Tabelle eine Paginierungssteuerung an (die Position kann standardmäßig mit dom und CSS verändert werden). Diese hat Knöpfe mit denen der Endnutzer zwischen den Seiten der Tabelle wechseln kann. Welche Knöpfe in der Paginierungssteuerung angezeigt werden, wird durch diese Option definiert.

DataTables hat sechs eingebaute Knopfeinstellungen:

- `numbers` - Nur Knöpfe für Seitennummer
- `simple` - Nur Knöpfe für 'Vorherige' und 'Nächste'
- `simple_numbers` - Knöpfe für 'Vorherige', 'Nächste' und Seitennummer
- `full` - Knöpfe für 'Erste', 'Vorherige', 'Nächste' und 'Letzte'
- `full_numbers` - Knöpfe für 'Erste', 'Vorherige', 'Nächste', 'Letzte' und Seitennummern
- `first_last_numbers` - Knöpfe für 'Erste', 'Letzte' und Seitennummern

Weitere Optionen können über Plugins hinzugefügt werden.

### 3.2.91 Beispiel

Nutze den Typ `full_numbers` der Paginierungskontrolle:

```
$('#example').dataTable( {
  "pagingType": "full_numbers"
} );
```

### 3.2.92 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- page()

#### Options

- dom
- paging

#### Erlaubte Werte

- numbers
- simple
- simple\_numbers
- full
- full\_numbers
- first\_last\_numbers

#### table.renderer

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

DataTables fügt einige komplexe Komponenten zur HTML-Seite hinzu, zum Beispiel die Paginierungssteuerung. Die Logik, die berechnet, welche Informationen angezeigt werden sollen (welche Knöpfe im Fall der Paginierungssteuerung), liegt im Kern von DataTables und verändert sich generell nicht. Wie die Knöpfe tatsächlich angezeigt werden ändert sich hingegen mit den Stilvorgaben der Seite. So können beispielsweise die Knöpfe der Paginierung als li-Elemente in einer ul-Liste oder einfach als Sammlung von a-Knöpfen dargestellt werden.

Die Fähigkeit, verschiedene Renderer zu verwenden, während die Kernlogik die gleiche bleibt, ist Grundlage der Art, wie DataTables Integrationsmöglichkeiten für CSS-Frameworks wie Bootstrap, Foundation und JQuery UI bietet. Das verwendete HTML wird verändert, um die Anforderungen der verschiedenen Frameworks zu erfüllen.

Dieser Parameter kontrolliert welche Renderer verwendet werden. Der gegebene Wert wird verwendet falls ein solcher Renderer existiert, sonst wird der standardmäßige Renderer verwendet. Weitere Renderer können durch Plug-ins hinzugefügt werden.

DataTables unterstützt momentan zwei verschiedene Renderer-Typen:

- header - Renderer der Kopfzeile
- pageButton - Renderer der Paginierungsknöpfe

Diese Liste wird in zukünftigen Versionen von DataTables wahrscheinlich deutlich erweitert.

### 3.2.93 Typen

#### string

Versucht einen Renderer mit diesem Namen für jeden Renderer-Typ zu verwenden. Falls für einen Renderer-Typ kein Renderer mit diesem Namen gefunden wird, wird der Standard-Renderer verwendet.

#### objekt

Spezifiziere bestimmte Renderer für die verschiedenen Typen. Siehe die Dokumentation oben für eine Liste der verfügbaren Typen.

### 3.2.94 Beispiele

Verwende den Bootstrap Plug-in Renderer:

```
$('#example').dataTable( {  
  renderer: "bootstrap"  
} );
```

Spezifiziere einen bestimmten Renderer für jede verfügbare Operation:

```
$('#example').dataTable( {  
  renderer: {  
    "header": "jqueryui",  
    "pageButton": "bootstrap"  
  }  
} );
```

### 3.2.95 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### Options

- pagingType

## table.retrieve

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ruft das DataTables-Objekt für den gegebenen Selektor ab. Es ist anzumerken, dass, wenn die Tabelle bereits initialisiert wurde, dieser Parameter DataTables dazu bringt, einfach das bereits aufgebaute Objekt zurückzugeben. Änderungen an dem Initialisierungsobjekt, dass an DataTables übergeben wurde, werden nicht beachtet (Diesen Parameter auf true zu setzen ist eine Bestätigung, dass dies Verstanden wurde!).

Die destroy Option kann verwendet werden, um eine Tabelle mit anderen Einstellungen neu zu initialisieren, falls nötig.

### 3.2.96 Beispiel

DataTables-Instanz einer existierenden Tabelle abrufen:

```
$(document).ready( function() {
  initTable();
  tableActions();
} );

function initTable () {
  return $('#example').dataTable( {
    "scrollY": "200px",
    "paginate": false,
    "retrieve": true
  } );
}

function tableActions () {
  var table = initTable();

  // perform API operations with `table`
  // ...
}
```

### 3.2.97 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

## API

- destroy()

## Options

- destroy

## table.rowId

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: DT\_RowId ]

Es ist in einer Tabelle oft nützlich ein id-Attribut auf jedem tr-Element für Zeilenselektion und Identifikation von Datenquellen zu haben, besonders wenn Events verwendet werden.

DataTables versucht anhand dem Wert dieser Option automatisch für jede Zeile einen id-Wert aus der Datenquelle zu lesen. Standardmäßig ist der Wert DT\_RowId, er kann aber auch auf jeden anderen Namen gesetzt werden. So wie columns.data kann er auch über die punktierte Objektnotation von Javascript aus einer JSON-Datenquelle gelesen werden (z.B. DT\_RowId: 'image.id').

Wenn kein id-Wert für eine Zeile gefunden wird, bleibt die id der Zeile ungesetzt.

Alle id-Werte der Zeilen in der Datenquelle sollten die HTML-Spezifikation für gültige Werte erfüllen:

Der Wert muss einzigartig unter allen IDs im Unterbaum des Elements sein und mindestens einen Buchstaben enthalten. Der Wert darf keine Leerzeichen enthalten.

Es kann auch die CSS 2.1 Spezifikation für einen Identifikator verwendet werden, welche restriktiver als die von HTML5 ist und maximale Kompatibilität mit jQuery bietet:

Identifikatoren (was Elementnamen, Klassen und IDs in Selektoren beinhaltet) können nur die Buchstaben [a-zA-Z0-9], die ISO-10646 Buchstaben U+00A0 und höher, sowie den Bindestrich (-) und den Unterstrich (\_) enthalten. Sie können nicht mit einer Zahl, zwei Bindestrichen, oder einem Bindestrich gefolgt von einer Zahl anfangen. Identifikatoren können auch maskierte Zeichen und jeden ISO-10646 Buchstaben als numerischen Code enthalten.

### 3.2.98 Beispiel

Nutze die Eigenschaft staffId als Datenquelle für die id der Zeilen:

```
$('#myTable').DataTable( {
  ajax: '/api/staff',
  rowId: 'staffId'
} );
```



### 3.2.99 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `row().id()`
- `rows().ids()`

Options

- `rowId`

#### table.scrollCollapse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Wenn vertikales (y) Scrollen durch die Verwendung von `scrolly` aktiviert ist, zwingt DataTables die Höhe des Viewports der Tabelle durchgehend auf den angegebenen Wert (nützlich für das Layout). Dies kann jedoch seltsam aussehen, wenn auf ein kleines Datenset gefiltert wird und die Fußzeile weiter unten "schweben bleibt". Dieser Parameter kann aktiviert werden, um DataTables den Viewport zusammenklappen zu lassen, wenn das Datenset in einen Viewport mit kleiner Höhe passt.

### 3.2.100 Beispiel

Aktiviere zusammenklappen des Viewports:

```
$('#example').dataTable( {
  scrolly: 300,
  scrollCollapse: true
} );
```

### 3.2.101 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- `scrolly`

## table.search.caseInsensitive

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Filterung Groß- und Kleinschreibung ignorieren soll.

### 3.2.102 Beispiel

Filtern unter beachtung der Groß- und Kleinschreibung:

```
$('#example').dataTable( {
  "search": {
    "caseInsensitive": false
  }
});
```

### 3.2.103 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- columns().search()

Options

- search
- searchCols
- search
- search.search
- search.regex
- search.smart

## table.search.regex

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Reguläre Ausdrücke können verwendet werden um fantastisch komplexe Filterterme zu konstruieren, es ist aber auch absolut gültig wenn Nutzer Zeichen wie \* in den Filter eingeben. Es muss also eine Entscheidung getroffen werden, ob Sonderzeichen von regulären Ausdrücken maskiert werden sollen, oder nicht. Diese Option kontrolliert die Fähigkeit, dies zu tun.

Einfach gesagt ist diese Option eine Flagge, die indiziert, ob Suchterme als reguläre Ausdrücke interpretiert werden sollen (true) oder nicht (false).

### 3.2.104 Beispiel

Behandle Sucheingaben als reguläre Ausdrücke:

```
$('#example').dataTable( {
  "search": {
    "regex": true
  }
} );
```

### 3.2.105 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- columns().search()

Options

- search
- searchCols
- search
- search.search
- search.caseInsensitive
- search.smart

table.search.smart

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Die eingebaute Filterung von DataTables ist "intelligent" insofern, dass sie die Eingabe des Nutzers in einzelne Wörter aufteilt und nach diesen Wörtern in beliebiger Position und Reihenfolge in der Tabelle sucht (statt einen reinen Vergleich der Strings durchzuführen).

Dies kann die Nützlichkeit der Filterfunktion zwar signifikant erhöhen, wird aber durch einen komplexen regulären Ausdruck durchgeführt, der nutzerdefinierte reguläre Ausdrücke beeinträchtigen kann, wenn solche eingegeben werden können (search.regex). Daher kann die intelligente Filterung durch diese Option deaktiviert werden.

### 3.2.106 Beispiel

Intelligente Filterung deaktivieren:

```
$('#example').dataTable( {
  "search": {
    "smart": false
  }
} );
```

### 3.2.107 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- columns().search()

Options

- search
- searchCols
- search
- search.search
- search.caseInsensitive
- search.regex

table.search.search

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Initialer Suchterm der für die Tabelle verwendet werden soll.

### 3.2.108 Beispiel

Setze einen initialen Filter:

```
$('#example').dataTable( {
  "search": {
    "search": "Fred"
  }
} );
```

### 3.2.109 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- columns().search()

Options

- search
- searchCols
- search
- search.caseInsensitive
- search.regex
- search.smart

table.searchCols

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Prinzipiell identisch zur search Option, aber in diesem Fall für einzelne Spalten statt für den globalen Filter. Diese Option definiert die Filterung, die initial auf die Tabelle angewendet werden soll.

Die Größe des Arrays muss gleich der Anzahl der Spalten sein, und jedes Element ein Objekt mit den Parametern search und escapeRegex (letzteres ist optional). 'null' wird auch akzeptiert, und führt zur standardmäßigen Filterung auf der Spalte.

### 3.2.110 Beispiel

Setze initiale Filterung auf zweiter und vierter Spalte:

```
$('#example').dataTable( {
  "searchCols": [
    null,
    { "search": "My filter" },
    null,
    { "search": "[0-9]", "escapeRegex": false }
  ]
});
```

### 3.2.111 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `column().search()`
- `columns().search()`

Options

- `search`
- `search`
- `columns.searchable`

`table.searchDelay`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Für den Wert null setzt DataTables die Drosselung automatisch auf einen dem Bearbeitungsmodus entsprechenden Wert:

- Sofort - Klientseitige Bearbeitung
- 400mS - Serverseitige Bearbeitung

### 3.2.112 Beschreibung

Die eingebaute globale Suche von DataTables (standardmäßig oben rechts von jeder Tabelle) durchsucht die Tabelle sofort bei jedem Tastendruck wenn klientseitige Bearbeitung verwendet wird, und reduziert die Suchfrequenz automatisch auf 400mS wenn serverseitige Bearbeitung verwendet wird. Diese (Drosselung der) Suchfrequenz kann mit dem searchDelay-Parameter für klient- und serverseitige Bearbeitung kontrolliert werden.

Die Suchfrequenz kontrollieren zu können hat mehrere Nutzen:

- Eine reduzierte Suchfrequenz kann die Rechenlast auf älteren Browsern und Computern reduzieren
- Die Tabelle während der Suche weniger oft neu zu malen verwirrt den Nutzer weniger
- Reduziere die Last auf dem Server bei serverseitiger Bearbeitung durch weniger Aufrufe
- Andersherum, erhöhe die Geschwindigkeit der Suche bei serverseitiger Bearbeitung durch herabsetzung der Drosselung von 400mS zu sofort (0)

Wie bei vielen Teilen von DataTables muss der Entwickler selbst über die korrekte Konfiguration für seinen Anwendungsfall entscheiden.

Der Wert von searchDelay ist in Millisekunden (mS) angegeben.

Es ist anzumerken, dass diese Option nur die eingebaute globale Suchbox von DataTables betrifft. Die API-Methoden search() und column.search() werden nicht durch sie beeinträchtigt. Sollen Aufrufe an diese Funktionen gedrosselt werden, so sollte die Methode \$.fn.dataTable.util.throttle() verwendet werden.

### 3.2.113 Beispiel

Setze eine 350mS Drosselung für die Suche:

```
$('#example').DataTable( {
  searchDelay: 350
} );
```

### 3.2.114 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- search

table.stateDuration

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 7200 ]

Dauer für die gespeicherte Zustandsinformationen gültig sind. Nach dem Ablauf dieser Zeitspanne wird der Zustand zum Standardwert zurückgesetzt.

Diese Option zeigt DataTables auch an, ob localStorage oder sessionStorage verwendet werden soll, um den Zustand zu speichern. Wenn der Wert auf -1 steht, wird sessionStorage verwendet, für 0 oder größer wird localStorage verwendet.

Der Unterschied zwischen den beiden Speicher-APIs ist, dass sessionStorage Daten nur für die momentane Session behält (also für das momentane Browserfenster). Für mehr Informationen über die zwei HTML APIs siehe die Mozilla Speicherdokumentation.

Es ist anzumerken, dass der Wert in Sekunden angegeben wird. Der Wert 0 ist ein *besonderer* Wert, da er anzeigt, dass der Zustand ohne Zeitlimit gespeichert und geladen werden kann.

### 3.2.115 Beispiele

Setze Gültigkeit des gespeicherten Zustands auf einen Tag:

```
$('#example').dataTable( {
  "stateSave": true,
  "stateDuration": 60 * 60 * 24
} );
```

Benutze sessionStorage:

```
$('#example').dataTable( {
  "stateSave": true,
  "stateDuration": -1
} );
```

### 3.2.116 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- stateSave



## table.stripeClasses

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Array aus CSS-Klassen die in dieser Reihenfolge auf die angezeigten Zeilen angewendet werden sollen. Das Array kann von beliebiger Länge sein. DataTables wendet die Klassen in der gegebenen Reihenfolge an und kehrt zum Anfang des Arrays zurück, wenn das Ende erreicht wurde.

Es ist anzumerken, dass diese Option ihren Standardwert aus den \$.fn.dataTable.ext.classes.stripe\* Optionen erhält (standardmäßig 'odd' und 'even').

### 3.2.117 Beispiel

Benutze drei Streifenklassen:

```
$('#example').dataTable( {
  "stripeClasses": [ 'strip1', 'strip2', 'strip3' ]
} );
```

## table.tabIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Standardmäßig ermöglicht DataTables Tastaturnavigation durch die Tabelle (Sortierung, Paginierung und Filterung) indem ein tabIndex-Attribut zu den entsprechenden Elementen hinzugefügt wird. So kann der Endnutzer durch die Steuerungen tabben und sie mit Enter aktivieren, was die Steuerung der Tabelle ohne Maus ermöglicht.

Der Standardmäßige `tabIndex` ist 0, wodurch der Tab dem Fluss des Dokuments folgt. Dies kann mit diesem Parameter überschrieben werden. Mit einem Wert von -1 kann die eingebaute Tastaturnavigation ausgeschaltet werden, was aus Gründen der Zugänglichkeit nicht empfohlen wird.

### 3.2.118 Beispiel

```
$('#example').dataTable( {
  "tabIndex": 1
} );
```

### 3.2.119 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ordering
- paging
- searching

`table.columnDef.t[].cellType`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "th" ]

Ändere den Zellentyp der für eine Spalte verwendet wird - entweder td-Zellen oder th-Zellen.

Dies kann nützlich sein, da th-Zellen semantische Bedeutung im Tabellenkörper haben, wodurch sie als Kopf einer Zeile gelten können (es sollte potentiell `columns.createdCell` verwendet werden, um `scope='row'` zu den th-Elementen hinzuzufügen).

### 3.2.120 Beispiel

Lasse die erste Spalte th-Zellen verwenden:

```
$('#example').dataTable( {
  "ajax": "json.txt",
  "columnDefs": [ {
    "targets": 0,
    "cellType": "th"
  } ]
} );
```

### 3.2.121 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax
- columns.createdCell

Erlaubte Werte

- td
- th

table.columnDef.t[].className

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option fügt einfach eine Klasse zu jeder Zelle einer Spalte hinzu, ungeachtet ob die Tabellenquelle DOM, Javascript oder Ajax ist. Dies kann für das styling von Spalten nützlich sein.

### 3.2.122 Beispiel

Mit className über columnDefs eine Klasse zu den Zellen der ersten Spalte hinzufügen:

```
$('#example').dataTable( {
  "columnDefs": [
    { className: "my_class", "targets": [ 0 ] }
  ]
} );
```

### table.columnDef.t[].contentPadding

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das erste, das über diese Option gesagt werden sollte, ist, dass sie im Allgemeinen nicht benötigt wird.

Sie kann jedoch in einzelnen Fällen nützlich sein. Wenn DataTables die Breiten der einzelnen Spalten berechnet, sucht es den längsten String jeder Spalte, konstruiert eine temporäre Tabelle, und liest von dieser die Breiten ab. Das Problem dabei ist, dass "mmm" deutlich breiter als "iiii" ist, das letztere aber der längere String ist - dadurch kann die Berechnung fehlschlagen (das ganze korrekt zu machen indem die Werte in ein DOM-Objekt gesteckt werden welches dann vermessen wird ist schrecklich langsam!). Als "workaround" wird daher diese Option angeboten. Sie hängt ihren Wert an den längsten String der Spalte an - also padding.

### 3.2.123 Beispiel

Setze padding des Inhalts für die letzte Spalte der Tabelle:

```
$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    { "contentPadding": "mmm" }
  ]
} );
```

### 3.2.124 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- autoWidth

### table.columnDef.t[].data

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Nimmt automatisch den Indexwert der Spalte ]

Diese Eigenschaft kann verwendet werden, um Daten aus einer beliebigen Datenquelle, die auch ein tief verschachteltes Objekt sein kann, zu lesen und zu schreiben. `data` kann in einer Reihe von verschiedenen Formen gegeben werden, welche das Verhalten der Eigenschaft wie unten dokumentiert verändern.

Der Wert, der für einen Datenpunkt bestimmt wird (durch diese Option und `columns.render`) wird von `DataTables` für die verlangten Daten verwendet, mit drei Sonderfällen:

- `undefined` - Identisch zu nicht vorhanden!
- `null` - Wenn `columns.render` verwendet wird, wird die originale Datenquelle der Zeile an die Renderfunktion übergeben. Gibt es keinen Renderer, so wird der Wert von `columns.defaultContent` verwendet. Wenn `columns.defaultContent` nicht definiert ist, wird ein leerer String verwendet. Für alle anderen Datentypen wird `null` verwendet.
- `function` - Die Funktion wird ausgeführt und der Rückgabewert verwendet. Seit `DataTables 1.10.1` wird die Funktion im gleichen Scope wie das Datenobjekt der Zeile ausgeführt. Das Ergebnis davon ist, dass eine Objektinstanz als Datenquelle einer Zeile verwendet werden kann.

Man beachte, dass `data` sowohl ein getter als auch ein setter ist. Wird nur Formatierung von Daten für die Ausgabe benötigt, so sollte wahrscheinlich `columns.render` verwendet werden, was nur ein getter ist, und daher deutlich einfacher zu verwenden!

Seit `DataTables 1.10.3` kann diese Option mit DOM-Daten verwendet werden, um `DataTables` zu informieren, wo die gelesenen Daten für jede Spalte einer Datenquelle hingeschrieben werden sollen. Standardmäßig speichert `DataTables` die Daten in einem Array, aber mit dieser Option können Namen von Objekteigenschaften bereitgestellt werden, welche die Struktur eines zu verwendenden Objekts beschreiben.

### 3.2.125 Types

#### integer

Wird als Array-Index für die Datenquelle behandelt. Dies ist der Standard, den `DataTables` verwendet (inkrementiert für jede Spalte).

#### string

Lese und schreibe eine Objekteigenschaft von und zu der Datenquelle. Es gibt drei "spezielle" Optionen, die im String verwendet werden können, um zu verändern, wie `DataTables` die Daten aus dem Quellobjekt liest:

- `"."` - Punktierter Javascript-Notation. So wie ein Punkt in Javascript verwendet wird, um aus geschachtelten Objekten zu lesen, kann er auch in `data` verwendet werden. Zum Beispiel: `browser.version` oder `browser.name`. Wenn der Name des Objektparameters einen Punkt enthält, kann er mit `\\` maskiert werden. Zum Beispiel `first\\.name`
- `"[]"` - Array Notation. `DataTables` kann automatisch Daten aus einer Array-Quelle zusammenführen, wobei die Daten mit dem zwischen den eckigen Klammern angegebenen String verknüpft werden. Zum Beispiel: `name[, ]` würde aus dem Quellarray eine mit Kommas getrennte Liste machen. Wird zwischen den Klammern kein String angegeben, so wird das originale Quellarray zurückgegeben.

- "()" - Funktionsnotation. An das Ende eines Parameters () anzufügen führt eine Funktion des gegebenen Namens aus. Zum Beispiel: `browser()` für eine simple Funktion auf der Datenquelle, `browser.version()` für eine Funktion in einer geschachtelten Eigenschaft, oder sogar `browser().version` für die Eigenschaft eines von der Funktion zurückgegebenen Objekts. Man beachte, dass die Funktionsnotation für die Verwendung in `render` statt in `data` empfohlen ist, da sie als `Renderer` deutlich einfacher zu benutzen ist.

## null

Benutze die originale Datenquelle für die Zeile anstatt Daten aus ihr zu ziehen. Diese Aktion beeinflusst zwei andere Initialisierungsoptionen:

- `columns.defaultContent` - Wenn `null` für `data` gegeben wird und `defaultContent` für die Spalte definiert ist, wird der Wert von `defaultContent` für die Zelle verwendet.
- `columns.render` - Wenn `null` für `data` gegeben wird und die Option `render` für die Spalte definiert ist, wird die gesamte Datenquelle für die Zeile für den `Renderer` verwendet.

## objekt

Verwende verschiedene Daten für die verschiedenen von `DataTables` verlangten Datentypen (`filter`, `display`, `type` oder `sort`). Die Namen der Eigenschaften des Objekts stehen für den jeweiligen Datentyp und die Werte können als `integer`, `string` oder Funktion nach den gleichen Regeln wie oben definiert werden. Man beachte, dass eine `_`-Option definiert werden *muss*. Dies ist die Standardoption, die verwendet wird, wenn kein Parameter für den verlangten Wert definiert wurde.

Ein mögliches Beispiel:

```
"data": {
  "_": "phone",
  "filter": "phone_filter",
  "display": "phone_display"
}
```

## funktion data( row, type, set, meta )

Die gegebene Funktion wird aufgerufen wann immer `DataTables` den Wert einer Zelle der Spalte setzen oder holen muss.

Diese Funktion kann mehrmals aufgerufen werden, da `DataTables` sie für jeden benötigten Datentyp aufruft - `sort`, `filter` und `display`.

Man beachte, dass `DataTables` die Funktion nur dann als *setter* für eine neu hinzugefügte Zeile aufruft, wenn die Daten der Zeile aus dem DOM gelesen werden (die Tabelle also durch eine vorpopulierte HTML-Tabelle initialisiert wird). Die Funktion wird nicht als *setter* aufgerufen, wenn die Datenquelle `Ajax` oder `Javascript` ist, da angenommen wird, dass die Daten bereits im benötigten Format vorliegen.

Parameter:

1. Name: `row`, Typ: `Array|Objekt`, Optional: Nein  
Die Daten der gesamten Zeile.
2. Name: `type`, Typ: `String|Objekt`, Optional: Nein  
Der für die Zelle verlangte Datentyp. Vordefinierte Typen sind `filter`, `display`, `type` und `sort`.
3. Name: `set`, Typ: `Beliebig`, Optional: Nein  
Zu setzender Wert wenn der `type`-Parameter auf `set` steht. Sonst `undefined`.
4. Name: `meta`, Typ: `Objekt`, Optional: Nein

**Seit 1.10.1:** Ein Objekt, welches zusätzliche Informationen über die abgefragte Zelle enthält. Dieses Objekt hat die folgenden Eigenschaften:

- row - Der Zeilenindex der abgefragten Zelle. Siehe `row().index()`.
- col - Der Spaltenindex der abgefragten Zelle. Siehe `column().index()`.
- settings - Das `DataTables.Settings`-Objekt für die betroffene Tabelle. Dieses kann verwendet werden um eine API-Instanz zu erhalten, falls nötig.

Rückgabewert:

Der Rückgabewert wird nicht verwendet, wenn 'set' als type gewählt wurde. Ansonsten wird der Rückgabewert für die abgefragten Daten verwendet.

### 3.2.126 Beispiele

Lese Tabellendaten aus Objekten:

```
// JSON-Struktur für jede Zeile in diesem Beispiel:
// {
//   "engine": {value},
//   "browser": {value},
//   "platform": {value},
//   "version": {value},
//   "grade": {value}
// }
$('#example').dataTable( {
  "ajaxSource": "sources/objects.txt",
  "columns": [
    { "data": "engine" },
    { "data": "browser" },
    { "data": "platform" },
    { "data": "version" },
    { "data": "grade" }
  ]
} );
```

Lese Informationen aus verschachtelten Objekten:

```
// JSON-Struktur für jede Zeile:
//  {
//    "engine": {value},
//    "browser": {value},
//    "platform": {
//      "inner": {value}
//    },
//    "details": [
//      {value}, {value}
//    ]
//  }
$('#example').dataTable( {
  "ajaxSource": "sources/deep.txt",
  "columns": [
    { "data": "engine" },
    { "data": "browser" },
    { "data": "platform.inner" },
    { "data": "details.0" },
    { "data": "details.1" }
  ]
} );
```

Lese eine Tabelle aus einer DOM-Quelle in Datenobjekte:

```
$(document).ready(function() {
  $('#example').DataTable({
    "columns": [
      { "data": "name" },
      { "data": "position" },
      { "data": "office" },
      { "data": "age" },
      { "data": "start_date" },
      { "data": "salary" }
    ]
  });
} );
```

Verwende data als Funktion, um verschiedene Informationen für Sortierung, Filterung und Anzeige bereitzustellen. In diesem Fall Währung (Preis):



```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": function ( row, type, val, meta ) {
      if (type === 'set') {
        row.price = val;
        // Store the computed display and filter values for efficiency
        row.price_display = val=="" ? "" : "$"+numberFormat(val);
        row.price_filter = val=="" ? "" : "$"+numberFormat(val)+" "+val;
        return;
      }
      else if (type === 'display') {
        return row.price_display;
      }
      else if (type === 'filter') {
        return row.price_filter;
      }
      // 'sort', 'type' and undefined all just use the integer
      return row.price;
    }
  } ]
} );

```

Verwendung des Standardinhalts:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": [ 0 ],
    "data": null,
    "defaultContent": "Click to edit"
  } ]
} );

```

Verwendung von Array-Notation - Ausgeben eines Arrays als Liste:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": [ 0 ],
    "data": "name[ , ]"
  } ]
} );

```

### 3.2.127 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.render

### table.columnDef.t[].defaultContent

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Oftmals soll eine Spalte statischen Inhalt haben, zum Beispiel einfache Knöpfe zum Editieren und / oder Löschen, mit zugewiesenen Events. Diese Option ist für solche Fälle gedacht - statische Inhalte für Spalten erstellen. Soll dynamischer Inhalt erstellt werden (also basierend auf anderen Daten in der Zeile), so sollte die Option `columns.render` verwendet werden.

Weiterhin kann diese Option beim Laden von JSON-Daten nützlich sein, da der hier gesetzte Wert verwendet wird, wenn der Wert im JSON für die Zelle null ist (dann kann hier zum Beispiel der Wert "Nicht verfügbar" definiert werden).

### 3.2.128 Beispiele

Zeige eine Informationsnachricht für ein Feld, das den Wert null oder undefined haben kann:

```

$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    {
      "data": "first_name", // kann null oder undefined sein
      "defaultContent": "<i>Not set</i>"
    }
  ]
} );

```

Zeige einen leeren String, wenn ein Feld null oder undefined ist:

```
$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    {
      "data": "office", // can be null or undefined
      "defaultContent": ""
    }
  ]
} );
```

Erstelle einen Editierungsknopf in der letzten Spalte mit columnDefs:

```
$('#example').dataTable( {
  "columnDefs": [
    {
      "data": null,
      "defaultContent": "<button>Edit</button>",
      "targets": -1
    }
  ]
} );
```

### 3.2.129 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.data
- columns.render

table.columnDef.t[].name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bei der Arbeit mit der API von DataTables will man oft einzelne Spalten adressieren können, um mit ihnen zu arbeiten (zum Beispiel um den numerischen Inhalt einer Spalte aufzusummieren). DataTables hat zwei grundlegende Arten, Spalten zu adressieren:

- Über den Spaltenindex (automatisch bei der Initialisierung der Tabelle zugewiesen)
- Über einen Namen (mit dieser Option zugewiesen!)

Dies kann das Arbeiten mit Spalten in der API sehr einfach machen. Soll zum Beispiel auf die Daten der Spalte namens location zugegriffen werden, würde man `table.column('location:name').data()` benutzen. Das Anhängen des Strings `":name"` zeigt DataTables, dass eine Auswahl über den Spaltennamen vorgenommen werden soll. Siehe die Dokumentation der API-Funktion `column()` für mehr Informationen über Spaltenselektoren.

### 3.2.130 Beispiel

Spaltennamen mit `columnDefs` setzen:

```
$('#example').dataTable( {
  "columnDefs": [
    { "name": "engine", "targets": 0 },
    { "name": "browser", "targets": 1 },
    { "name": "platform", "targets": 2 },
    { "name": "version", "targets": 3 },
    { "name": "grade", "targets": 4 }
  ]
} );
```

### 3.2.131 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `column()`
- `columns()`
- `column().index()`
- `columns().indexes()`

`table.columnDef.t[].orderable`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Mit diesem Parameter kann dem Endnutzer die Fähigkeit entzogen werden, eine Spalte zu sortieren. Dies kann für Spalten mit generiertem Inhalt nützlich sein, zum Beispiel wenn die Tabelle Knöpfe zum Löschen oder Editieren enthält.

Man beachte, dass diese Option nur die Sortierfähigkeit des Endnutzers beeinträchtigt. Entwickler können die Spalte weiterhin über die Option `order` oder die API-Methode `order()` sortieren.

### 3.2.132 Beispiel

Sortieren der ersten Spalte verbieten mit `columnDefs`:

```
$('#example').dataTable( {
  "columnDefs": [
    { "orderable": false, "targets": 0 }
  ]
} );
```

### 3.2.133 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `order()`
- `column().order()`
- `columns().order()`

Options

- `ordering`
- `order`

`table.columnDef.t[].orderData`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Erlaubt der Sortierung einer Spalte die Daten, nach denen sortiert wird, entweder aus einer anderen (oftmals verborgenen) Spalte zu holen, oder aus mehreren Spalten.

Ein häufiges Beispiel ist eine Tabelle, welche Spalten für Vor- und Nachnamen nebeneinander enthält. Es ist intuitiv, dass über diese eine Mehrspaltige Sortierung angewendet werden würde. Ein weiteres Beispiel, mit einer einzelnen Zeile, ist der Fall, dass die dem Nutzer angezeigten Daten selber nicht sortierbar sind (eine Spalte mit Bildern), es aber sortierbare Metadaten gibt (z.B. Dateinamen). Man beachte, dass für diesen Zweck alternativ orthogonale Daten verwendet werden können.

Der Wert muss ein Spaltenindex oder ein Array aus Spaltenindices sein.

### 3.2.134 Beispiel

Benutze `orderData` um eine Sortierung über mehrere Spalten mit `columnDefs` zu definieren:

```
$('#example').dataTable( {
  "columnDefs": [
    { "orderData": [ 0, 1 ], "targets": 0 },
    { "orderData": 0, "targets": 1 },
    { "orderData": [ 2, 3, 4 ], "targets": 2 }
  ]
} );
```

### 3.2.135 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `order()`
- `column().order()`
- `columns().order()`

Options

- `ordering`
- `order`

`table.columnDef.t[].orderDataType`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die primäre Sortiermethode von DataTables (die Option `ordering`) verwendet aus Performanzgründen Daten, die im Speichercache liegen, anstatt die Daten bei jeder Anfrage direkt aus dem DOM zu lesen (was von Natur aus langsam ist). Es gibt jedoch Fälle, in denen trotz der schlechteren Performanz direkt aus dem DOM gelesen werden soll, zum

Beispiel wenn die Tabelle Formularelemente enthält, deren Werte durch den Nutzer verändert werden können. Diese Konfigurationsoption wird bereitgestellt, um Plugins das Anbieten dieser Fähigkeit zu erlauben.

Man beachte, dass keine `columns.orderDataType` Plugins in `DataTables` eingebaut sind. Sie müssen separat hinzugefügt werden. Siehe die Seite der Sortierplugins für `DataTables` für weitere Informationen.

### 3.2.136 Beispiel

Setze Typ der Sortierungsdaten mit `columnDefs`:

```
$('#example').dataTable( {
  "columnDefs": [
    { "orderDataType": "dom-text", "targets": [ 2, 3 ] },
    { "type": "numeric", "targets": 3 },
    { "orderDataType": "dom-select", "targets": 4 },
    { "orderDataType": "dom-checkbox", "targets": 5 }
  ]
} );
```

### 3.2.137 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- `ordering`
- `columns.type`

`table.columnDef.t[].orderSequence`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [ 'asc', 'desc' ] ]

Die standardmäßige Sortierrichtung sowie das Verhalten des Sortierleiters (z.B. nur aufsteigendes Sortieren erlauben) kann mit diesem Parameter kontrolliert werden.

### 3.2.138 Beispiel

Definiere die angewandte Sortierrichtung mit `orderSequence` über `columnDefs`:

```

$('#example').dataTable( {
  "columnDefs": [
    { "orderSequence": [ "asc" ], "targets": [ 1 ] },
    { "orderSequence": [ "desc", "asc", "asc" ], "targets": [ 2 ] },
    { "orderSequence": [ "desc" ], "targets": [ 3 ] }
  ]
} );

```

### 3.2.139 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `order()`
- `column().order()`
- `columns().order()`

#### Options

- `ordering`
- `order`

#### Erlaubte Werte

- `asc`
- `desc`

#### table.columnDef.t[].render

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Eigenschaft modifiziert die Daten, die von DataTables für verschiedene Operationen benutzt werden, wenn sie aus der Datenquelle gelesen werden. `columns.render` kann als nur lesende Partneroption zu `columns.data` betrachtet werden, wobei `columns.data` lesen und schreiben kann (und daher komplexer ist). Wie `columns.data` kann diese Option auf verschiedene Arten gegeben werden, um ihr Verhalten wie unten beschrieben zu beeinflussen.

Die von `columns.render` zurückgegebenen Daten (egal ob die Option als Funktion, Integer oder String verwendet wird) werden von DataTables für den verlangten Datentyp verwendet (man nennt sie auch die *aufgelösten Daten*). Die Fähigkeit von DataTables, verschiedene Daten für seine verschiedenen Operationen zu verlangen, wird bezeichnet als "orthogonale Daten", und ermöglicht die Verwendung von verschiedenen Formen der gleichen



Daten für verschiedene Operationen (zum Beispiel kann ein Datum formatiert sein, um es dem Endnutzer anzuzeigen, aber intern als Integer zur Sortierung benutzt werden).

Es gibt drei spezielle Werte, zu denen `columns.render` aufgelöst werden kann:

- **Undefiniert** - Der Wert von `columns.defaultContent` wird verwendet. Ist dieser nicht spezifiziert, so wird ein Fehler gegeben.
- **null** - Der Wert von `columns.defaultContent` wird verwendet. Ist dieser nicht definiert, so wird für angezeigte Daten ein leerer String verwendet. Für alle anderen Datentypen wird null verwendet.
- **Funktion** - Die Funktion wird ausgeführt und der Rückgabewert verwendet. Seit DataTables 1.10.1 wird die Funktion im gleichen Scope wie das Datenobjekt der Zeile ausgeführt. Dies führt dazu, dass eine Objektinstanz als Datenquelle einer Zeile verwendet werden kann. Siehe die unteren Beispiele für die Verwendung einer Instanz mit DataTables.

## Array zugriff

Wenn diese Option als String verwendet wird (siehe unten), kann man durch sie sowohl mehrere Datenpunkte, als auch einen einzelnen spezifischen Wert aus einem Array erhalten. Es ist wichtig den Unterschied zwischen den beiden Formen zu verstehen, wenn auf Arraydaten im Datenobjekt der Zeile zugegriffen werden soll.

- Einen oder mehr Werte aus einem Array erhalten - Verwende `[]`-Notation
- Ein einzelnes Element aus dem Array erhalten - Verwende Punktnotation - z.B. `".0"` um auf das erste Element des Arrays zuzugreifen.

Man betrachte zum Beispiel die folgende Datenstruktur:

```
"access": [
  { "id": "1", "name": "Printer" },
  { "id": "3", "name": "Desktop" },
  { "id": "4", "name": "VMs" }
]
```

Um die name-Eigenschaft der Objekte im Array anzuzeigen, in einer einzelnen Zelle, verwende `"access[,].name"` - dadurch werden die Namen durch Kommas getrennt aneinandergehängt. Das Ergebnis wäre hier also "Printer, Desktop, VMs".

Um auf ein einzelnes Attribut in einem Array zuzugreifen, verwende `".{index}"`. Soll aus dem obigen JSON nur der Name des ersten Elements gelesen werden, würde man also `"access.0.name"` verwenden. Das Ergebnis wäre hier "Printer".

## 3.2.140 Typen

### integer

Wird als Array-Index der Datenquelle behandelt. So wird die Option standardmäßig von DataTables verwendet (für jede Spalte inkrementiert).

### string

Lese ein Objekt aus der Datenquelle. Es gibt drei "spezielle" Optionen, die im String verwendet werden können, um zu verändern, wie DataTables die Daten aus dem Quellobjekt liest:

- `.-` - Punktierter Javascript-Notation. So wie ein Punkt in Javascript verwendet wird, um aus geschachtelten Objekten zu lesen, kann er auch in data verwendet werden. Zum Beispiel: `browser.version` oder

browser.name. Wenn der Name des Objektparameters einen Punkt enthält, kann er mit `\\` maskiert werden. Zum Beispiel `first\\.name`

- `[]` - Array Notation. DataTables kann automatisch Daten aus einer Array-Quelle zusammenführen, wobei die Daten mit dem zwischen den eckigen Klammern angegebenen String verknüpft werden. Zum Beispiel: `name[, ]` würde aus dem Quellarray eine mit Kommas getrennte Liste machen. Wird zwischen den Klammern kein String angegeben, so wird das originale Quellarray zurückgegeben.
- `()` - Funktionsnotation. An das Ende eines Parameters `()` anzufügen führt eine Funktion des gegebenen Namens aus. Zum Beispiel: `browser()` für eine simple Funktion auf der Datenquelle, `browser.version()` für eine Funktion in einer geschachtelten Eigenschaft, oder sogar `browser().version` für die Eigenschaft eines von der Funktion zurückgegebenen Objekts. Man beachte, dass die Funktionsnotation für die Verwendung in `columns.render` statt in `columns.data` empfohlen ist, da sie als Renderer deutlich einfacher zu benutzen ist.

## objekt

Verwende verschiedene Daten für die verschiedenen von DataTables verlangten Datentypen (`filter`, `display`, `type` oder `sort`). Die Namen der Eigenschaften des Objekts stehen für den jeweiligen Datentyp und die Werte können als `integer`, `string` oder Funktion nach den gleichen Regeln wie sonst in `columns.render` definiert werden.

Man beachte, dass eine Optionale `"_"`-Eigenschaft spezifiziert werden kann. Diese definiert den Standardwert, der verwendet wird, wenn für den von DataTables verlangten Datentyp keine Eigenschaft definiert ist. Ist für den verlangten Datentyp keine Eigenschaft definiert und die `"_"`-Eigenschaft wurde nicht spezifiziert, so wird der Datenwert verwendet, auf den `columns.data` zeigt.

Ein mögliches Beispiel:

```
"data": "phone",
"render": {
  "_": "plain",
  "filter": "filter",
  "display": "display"
}
```

## funktion render( data, type, row, meta )

Die gegebene Funktion wird aufgerufen wann immer DataTables den Wert einer Zelle der Spalte holen muss. Man beachte, dass diese Funktion mehrmals aufgerufen werden kann, da DataTables sie für jeden benötigten Datentyp aufruft - `sort`, `filter` und `display`.

Parameter:

1. Name: `data`, Typ: Beliebig, Optional: Nein  
Die Daten der Zelle (basierend auf `columns.data`).
2. Name: `type`, Typ: String, Optional: Nein  
Der verlangte Datentyp. Dieses Attribut wird für die Unterstützung von orthogonalen Daten in DataTables verwendet. Der Wert ist einer der folgenden:
  - `"filter"`: Hole den Wert, den DataTables für die Filterung auf dieser Zelle verwenden soll.
  - `"display"`: Der Wert, der in der Tabelle angezeigt werden soll.
  - `"type"`: Der Wert, der für Typerkennung verwendet werden soll. Dieser sollte normalerweise mit dem Wert von `sort` übereinstimmen - siehe die Dokumentation des Typerkennungs-Plugins.
  - Undefined: Verwende die originalen (unmodifizierten) Daten für die Zelle.
  - Eigener Wert: Es ist möglich für Plugins wie Responsive (durch die Option `responsive.orthogonal`) oder Buttons (`buttons.exportData()`), vom vom Entwickler definierte Datentypen zu verlangen. Dies kann nützlich sein, wenn bestimmte Daten an eine spezifische Erweiterung geschickt werden sollen.

Siehe auch die Methode `cell().render()`, welche verwendet werden kann, um die gegebene Rendermethode zu beliebigen Zeitpunkten auszuführen.

3. Name: `row`, Typ: Beliebig, Optional: Nein

Die komplette Datenquelle der Zeile (nicht basierend auf `columns.data`).

4. Name: `meta`, Typ: Objekt, Optional: Nein

**Seit 1.10.1:** Ein Objekt, welches zusätzliche Informationen über die abgefragte Zelle enthält. Dieses Objekt hat die folgenden Eigenschaften:

- `row` - Der Zeilenindex der abgefragten Zelle. Siehe `row().index()`.
- `col` - Der Spaltenindex der abgefragten Zelle. Siehe `column().index()`.
- `settings` - Das `DataTables.Settings`-Objekt für die betroffene Tabelle. Dieses kann verwendet werden um eine API-Instanz zu erhalten, falls nötig.

Rückgabewert:

Der Rückgabewert der Funktion wird für die verlangten Daten verwendet.

### 3.2.141 Beispiele

Eine durch Kommas getrennte Liste aus einem Array aus Objekten erstellen:

```
$('#example').dataTable( {  
  "ajaxSource": "sources/deep.txt",  
  "columns": [  
    { "data": "engine" },  
    { "data": "browser" },  
    {  
      "data": "platform",  
      "render": "[, ].name"  
    }  
  ]  
});
```

Als Objekt, extrahiere verschiedene Daten für die verschiedenen Typen:

```
// Dies würde benutzt werden mit einer Datenquelle der Form:
//   { "phone": 5552368, "phone_filter": "5552368 555-2368", "phone_display":
//     "555-2368", ... }
// Hier wird der `phone`-Integer für Sortierung und Typerkennung verwendet, während
// `phone_filter`
// (welches beide Formen enthält) für die Filterung verwendet wird, wenn der Nutzer
// eine der beiden Formen eingibt, während
// die formatierte Telefonnummer in der Tabelle angezeigt wird.
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": null, // Nutze das ganze Quellobjekt als Quelle des Renderers
    "render": {
      "_": "phone",
      "filter": "phone_filter",
      "display": "phone_display"
    }
  } ]
} );
```

Wie oben, mit der 'phone'-Information als Objekt:

```
// Dies würde benutzt werden mit einer Datenquelle der Form:
//   "phone": { "plain": 5552368, "filter": "5552368 555-2368", "display":
//     "555-2368" }
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": 'phone',
    "render": {
      "_": "plain",
      "filter": "filter",
      "display": "display"
    }
  } ]
} );
```

Verwende eine Funktion um einen Link aus der Datenquelle zu konstruieren:

```
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": "download_link",
    "render": function ( data, type, row, meta ) {
      return '<a href="'+data+'">Download</a>';
    }
  } ]
} );
```

Zeige Punkte für lange Strings:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 4,
    "data": "description",
    "render": function ( data, type, row, meta ) {
      return type === 'display' && data.length > 40 ?
        '<span title="'+data+'"'>'+data.substr( 0, 38 )+'...</span>' :
        data;
    }
  } ]
} );

```

Eine Objektinstanz als Datenquelle der Zeile verwenden:

```

function Person( name, age, position ) {
  this._name    = name;
  this._age     = age;
  this._position = position;

  this.name = function () {
    return this._name;
  };

  this.age = function () {
    return this._age;
  };

  this.position = function () {
    return this._position;
  };
}

$(document).ready(function() {
  var table = $('#example').DataTable({
    columns: [
      { data: null, render: 'name' },
      { data: null, render: 'age' },
      { data: null, render: 'position' }
    ]
  });

  table.row.add( new Person( 'Airi Satou',    33, 'Accountant' ) );
  table.row.add( new Person( 'Angelica Ramos', 47, 'Chief Executive Officer
(CEO)' ) );
  table.row.add( new Person( 'Ashton Cox',    66, 'Junior Technical Author' ) );
  table.draw();
} );

```

### 3.2.142 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.data

table.columnDef.t[].searchable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Mit diesem Parameter kann festgelegt werden, ob DataTables diese Spalte in die filterbaren Daten der Tabelle miteinbezieht. Es kann zum Beispiel Sinn machen, diese Option für generierte Spalten, wie Knöpfe zum Editieren und Löschen, zu deaktivieren.

### 3.2.143 Beispiel

Suche auf der ersten Spalte mit columnDefs deaktivieren:

```
$('#example').dataTable( {
  "columnDefs": [
    { "searchable": false, "targets": 0 }
  ]
} );
```

### 3.2.144 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- column().search()
- columns().search()

Options

- search

## table.columnDef.t[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Wert gelesen aus der Kopfzeile der Spalte. ]

Die Spaltentitel werden normalerweise direkt aus dem DOM gelesen (von den Zellen im THEAD-Element), es kann aber oft nützlich sein, entweder den bestehenden Wert zu überschreiben, oder DataTables tatsächlich einen Header mit Spaltentiteln erstellen zu lassen (zum Beispiel wenn die Tabelle vor der Konstruktion von DataTables kein THEAD-Element hat). Diese Option bietet diese Möglichkeit.

Man beachte, dass DataTables nur einen einfachen Header mit einer einzelnen Zelle pro Spalte erstellen kann. Komplexe Header mit colspan und rowspan Attributen müssen entweder bereits im Dokument definiert sein, oder mit den normalen DOM / jQuery Methoden konstruiert werden.

### 3.2.145 Beispiel

Setze den Titel der ersten Spalte mit columnDefs:

```
$('#example').dataTable( {
  "columnDefs": [
    { "title": "My column title", "targets": 0 }
  ]
} );
```

## table.columnDef.t[].type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bei der Ausführung im Modus klientseitiger Verarbeitung kann DataTables die Daten, die für die Anzeige in den Zellen benutzt werden, auf eine zur auszuführenden Aktion passenden Art verarbeiten. Zum Beispiel werden HTML-Tags aus den Strings entfernt, die für die Filtersuche benutzt werden, während die Sortierung potentiell Währungssymbole entfernt, um Geldwerte numerisch Sortieren zu können. Die ausgeführte Formatierungsaktion, die die Daten *normalisiert*, damit sie für Sortierung und Suche verwendbar werden, hängt vom *Typ* der Spalte ab.

DataTables hat eine Reihe von eingebauten Typen, die automatisch erkannt werden:

- **date:** Datums- / Zeitwerte. Man beachte, dass das von DataTables eingebaute Datumsparse die Javascript-Methode `Date.parse()` verwendet, welche nur eine begrenzte Menge an Daten unterstützt. Zusätzliche Unterstützung für Datumsformatierung kann durch Plugins hinzugefügt werden.
  - Sortierung - Chronologisch sortiert
  - Filterung - Kein Effekt
- **num:** Simple Zahlensortierung.
  - Sortierung - Numerisch sortiert
  - Filterung - Kein Effekt
- **num-fmt:** Numerisches Sortieren von formatierten Zahlen. Zahlen mit Tausendertrennzeichen, Währungssymbolen oder Prozentzeichen werden von DataTables automatisch numerisch sortiert.
  - Unterstützte eingebaute Währungssymbole sind \$, £, € und ¥.
  - Unterstützte eingebaute Tausendertrennzeichen sind ' und ,
  - Beispiele:
    - \$100,000 - Sortiert als 100000
    - £10'000 - Sortiert als 10000
    - 5'000 - Sortiert als 5000
    - 40% - Sortiert als 40
  - Sortierung - Numerisch sortiert
  - Filterung - Kein Effekt
- **html-num:** Wie num, nur mit zusätzlichen HTML-Tags in den Daten.
  - Sortierung - Numerisch sortiert
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **html-num-fmt:** wie num-fmt, nur mit zusätzlichen HTML-Tags in den Daten.
  - Sortierung - Numerisch sortiert
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **html:** Einfache Verarbeitung von Strings mit HTML-Tags.
  - Sortierung - Sortiert mit entfernten HTML-Tags
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **string:** Rückfalltyp, falls die Daten der Spalte nicht zu den Anforderungen von einem der obigen Typen passen.
  - Sortierung - Kein Effekt
  - Filterung - Kein Effekt

Es wird erwartet, dass die obigen Optionen die meisten von DataTables verwendeten Datentypen abdecken, da Daten aber flexibel sind und in vielen Formen kommen, können zusätzliche Typen mit anderen Effekten durch Plugins hinzugefügt werden. So kann fast jedes vorstellbare Datenformat sortiert werden!

Als Optimierung kann der Spaltentyp, wenn er schon im voraus bekannt ist, über diesen Parameter gesetzt werden. Das erspart DataTables das Ausführen der Erkennungsroutine.

Es ist zu beachten, dass dieser Parameter bei serverseitiger Bearbeitung (`serverSide`) keinen Effekt hat, da Sortierung und Filterung auf dem Server durchgeführt werden.



### 3.2.146 Beispiel

Setze manuell den Typ der ersten Spalte mit columnDefs:

```
$('#example').dataTable( {
  "columnDefs": [
    { "type": "html", "targets": 0 }
  ]
} );
```

### 3.2.147 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ordering
- searching

table.columnDef.t[].visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

DataTables kann durch diese Option und die API-Methoden `column().visible()` / `columns().visible()` Spalten dynamisch anzeigen und verbergen. Diese Option kann verwendet werden, um den initialen Zustand der Sichtbarkeit der Spalte zu setzen, während die API-Methoden den Zustand zu einem späteren Zeitpunkt verändern können.

Dies ist besonders nützlich wenn die Tabelle eine große Anzahl Spalten hat, da so dem Nutzer die Möglichkeit gegeben werden kann, selber zu entscheiden, welche Spalten angezeigt werden sollen. Alternativ können mit dieser Option Spalten verbergen werden, die der Nutzer nicht sehen soll (zum Beispiel eine Spalte mit IDs einer Datenbank).

### 3.2.148 Beispiel

Die erste Spalte mit columnDefs verbergen:

```
$('#example').dataTable( {
  "columnDefs": [
    { "visible": false, "targets": 0 }
  ]
} );
```

### 3.2.149 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `column().visible()`
- `columns().visible()`

#### Options

- `scrollX`

#### table.columnDef.t[].width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dieser Parameter kann verwendet werden, um die Breite einer Spalte zu definieren, und kann jeden CSS-Wert akzeptieren (3em, 20px, etc.).

Man beachte, dass auf den Pixel genaue Spaltenbreite in Tabellen mit dynamischem Inhalt praktisch unmöglich ist, weshalb es nicht verwunderlich ist, wenn die tatsächliche Breite der Spalte um ein paar Pixel von dem hier definierten Wert abweicht. Die Spaltenbreite in der Tabelle hängt von vielen Aspekten wie den Zellenrändern, den Tabellenrändern, der border-collapse-Eigenschaft, dem Inhalt und anderem ab. Sowohl DataTables als auch der Browser versuchen unter Betrachtung all dieser Aspekte ein optimales Layout für die Tabelle zu finden.

### 3.2.150 Beispiel

Setze über `columnDefs` die Breite der ersten Spalte auf 20%:

```
$('#example').dataTable( {
  "columnDefs": [
    { "width": "20%", "targets": 0 }
  ]
} );
```

### 3.2.151 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `columns.adjust()`

Options

- `autoWidth`

`table.columns.i[].cellType`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "th" ]

Ändere den Zellentyp der für eine Spalte verwendet wird - entweder td-Zellen oder th-Zellen.

Dies kann nützlich sein, da th-Zellen semantische Bedeutung im Tabellenkörper haben, wodurch sie als Kopf einer Zeile gelten können (es sollte potentiell `columns.createdCell` verwendet werden, um `scope='row'` zu den th-Elementen hinzuzufügen).

### 3.2.152 Beispiel

Lasse die erste Spalte th-Zellen verwenden:

```
$('#example').dataTable( {
  "ajax": "json.txt",
  "columnDefs": [ {
    "targets": 0,
    "cellType": "th"
  } ]
} );
```

### 3.2.153 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax
- columns.createdCell

Erlaubte Werte

- td
- th

table.columns.i[].className

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option fügt einfach eine Klasse zu jeder Zelle einer Spalte hinzu, ungeachtet ob die Tabellenquelle DOM, Javascript oder Ajax ist. Dies kann für das styling von Spalten nützlich sein.

### 3.2.154 Beispiel

Mit className über columns eine Klasse zu den Zellen der ersten Spalte hinzufügen:

```
$('#example').dataTable( {
  "columns": [
    { className: "my_class" },
    null,
    null,
    null,
    null
  ]
} );
```

### table.columns.i[].contentPadding

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das erste, das über diese Option gesagt werden sollte, ist, dass sie im Allgemeinen nicht benötigt wird.

Sie kann jedoch in einzelnen Fällen nützlich sein. Wenn DataTables die Breiten der einzelnen Spalten berechnet, sucht es den längsten String jeder Spalte, konstruiert eine temporäre Tabelle, und liest von dieser die Breiten ab. Das Problem dabei ist, dass "mmm" deutlich breiter als "iiii" ist, das letztere aber der längere String ist - dadurch kann die Berechnung fehlschlagen (das ganze korrekt zu machen indem die Werte in ein DOM-Objekt gesteckt werden welches dann vermessen wird ist schrecklich langsam!). Als "workaround" wird daher diese Option angeboten. Sie hängt ihren Wert an den längsten String der Spalte an - also padding.

### 3.2.155 Beispiel

Setze padding des Inhalts für die letzte Spalte der Tabelle:

```
$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    { "contentPadding": "mmm" }
  ]
});
```

### 3.2.156 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- autoWidth

### table.columns.i[].data

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Nimmt automatisch den Indexwert der Spalte ]

Diese Eigenschaft kann verwendet werden, um Daten aus einer beliebigen Datenquelle, die auch ein tief verschachteltes Objekt sein kann, zu lesen und zu schreiben. `data` kann in einer Reihe von verschiedenen Formen gegeben werden, welche das Verhalten der Eigenschaft wie unten dokumentiert verändern.

Der Wert, der für einen Datenpunkt bestimmt wird (durch diese Option und `columns.render`) wird von `DataTables` für die verlangten Daten verwendet, mit drei Sonderfällen:

- `undefined` - Identisch zu nicht vorhanden!
- `null` - Wenn `columns.render` verwendet wird, wird die originale Datenquelle der Zeile an die Renderfunktion übergeben. Gibt es keinen Renderer, so wird der Wert von `columns.defaultContent` verwendet. Wenn `columns.defaultContent` nicht definiert ist, wird ein leerer String verwendet. Für alle anderen Datentypen wird `null` verwendet.
- `function` - Die Funktion wird ausgeführt und der Rückgabewert verwendet. Seit `DataTables 1.10.1` wird die Funktion im gleichen Scope wie das Datenobjekt der Zeile ausgeführt. Das Ergebnis davon ist, dass eine Objektinstanz als Datenquelle einer Zeile verwendet werden kann.

Man beachte, dass `data` sowohl ein getter als auch ein setter ist. Wird nur Formatierung von Daten für die Ausgabe benötigt, so sollte wahrscheinlich `columns.render` verwendet werden, was nur ein getter ist, und daher deutlich einfacher zu verwenden!

Seit `DataTables 1.10.3` kann diese Option mit DOM-Daten verwendet werden, um `DataTables` zu informieren, wo die gelesenen Daten für jede Spalte einer Datenquelle hingeschrieben werden sollen. Standardmäßig speichert `DataTables` die Daten in einem Array, aber mit dieser Option können Namen von Objekteigenschaften bereitgestellt werden, welche die Struktur eines zu verwendenden Objekts beschreiben.

### 3.2.157 Types

#### integer

Wird als Array-Index für die Datenquelle behandelt. Dies ist der Standard, den `DataTables` verwendet (inkrementiert für jede Spalte).

#### string

Lese und schreibe eine Objekteigenschaft von und zu der Datenquelle. Es gibt drei "spezielle" Optionen, die im String verwendet werden können, um zu verändern, wie `DataTables` die Daten aus dem Quellobjekt liest:

- `"."` - Punktierter Javascript-Notation. So wie ein Punkt in Javascript verwendet wird, um aus geschachtelten Objekten zu lesen, kann er auch in `data` verwendet werden. Zum Beispiel: `browser.version` oder `browser.name`. Wenn der Name des Objektparameters einen Punkt enthält, kann er mit `\\` maskiert werden. Zum Beispiel `first\\.name`
- `"[]"` - Array Notation. `DataTables` kann automatisch Daten aus einer Array-Quelle zusammenführen, wobei die Daten mit dem zwischen den eckigen Klammern angegebenen String verknüpft werden. Zum Beispiel: `name[, ]` würde aus dem Quellarray eine mit Kommas getrennte Liste machen. Wird zwischen den Klammern kein String angegeben, so wird das originale Quellarray zurückgegeben.

- "()" - Funktionsnotation. An das Ende eines Parameters () anfügen führt eine Funktion des gegebenen Namens aus. Zum Beispiel: `browser()` für eine simple Funktion auf der Datenquelle, `browser.version()` für eine Funktion in einer geschachtelten Eigenschaft, oder sogar `browser().version` für die Eigenschaft eines von der Funktion zurückgegebenen Objekts. Man beachte, dass die Funktionsnotation für die Verwendung in `render` statt in `data` empfohlen ist, da sie als `Renderer` deutlich einfacher zu benutzen ist.

## null

Benutze die originale Datenquelle für die Zeile anstatt Daten aus ihr zu ziehen. Diese Aktion beeinflusst zwei andere Initialisierungsoptionen:

- `columns.defaultContent` - Wenn `null` für `data` gegeben wird und `defaultContent` für die Spalte definiert ist, wird der Wert von `defaultContent` für die Zelle verwendet.
- `columns.render` - Wenn `null` für `data` gegeben wird und die Option `render` für die Spalte definiert ist, wird die gesamte Datenquelle für die Zeile für den `Renderer` verwendet.

## objekt

Verwende verschiedene Daten für die verschiedenen von `DataTables` verlangten Datentypen (`filter`, `display`, `type` oder `sort`). Die Namen der Eigenschaften des Objekts stehen für den jeweiligen Datentyp und die Werte können als `integer`, `string` oder Funktion nach den gleichen Regeln wie oben definiert werden. Man beachte, dass eine `_`-Option definiert werden *muss*. Dies ist die Standardoption, die verwendet wird, wenn kein Parameter für den verlangten Wert definiert wurde.

Ein mögliches Beispiel:

```
"data": {
  "_": "phone",
  "filter": "phone_filter",
  "display": "phone_display"
}
```

## funktion data( row, type, set, meta )

Die gegebene Funktion wird aufgerufen wann immer `DataTables` den Wert einer Zelle der Spalte setzen oder holen muss.

Diese Funktion kann mehrmals aufgerufen werden, da `DataTables` sie für jeden benötigten Datentyp aufruft - `sort`, `filter` und `display`.

Man beachte, dass `DataTables` die Funktion nur dann als *setter* für eine neu hinzugefügte Zeile aufruft, wenn die Daten der Zeile aus dem DOM gelesen werden (die Tabelle also durch eine vorpopulierte HTML-Tabelle initialisiert wird). Die Funktion wird nicht als *setter* aufgerufen, wenn die Datenquelle `Ajax` oder `Javascript` ist, da angenommen wird, dass die Daten bereits im benötigten Format vorliegen.

Parameter:

1. Name: `row`, Typ: `Array|Objekt`, Optional: Nein  
Die Daten der gesamten Zeile.
2. Name: `type`, Typ: `String|Objekt`, Optional: Nein  
Der für die Zelle verlangte Datentyp. Vordefinierte Typen sind `filter`, `display`, `type` und `sort`.
3. Name: `set`, Typ: `Beliebig`, Optional: Nein  
Zu setzender Wert wenn der `type`-Parameter auf `set` steht. Sonst `undefined`.
4. Name: `meta`, Typ: `Objekt`, Optional: Nein

**Seit 1.10.1:** Ein Objekt, welches zusätzliche Informationen über die abgefragte Zelle enthält. Dieses Objekt hat die folgenden Eigenschaften:

- row - Der Zeilenindex der abgefragten Zelle. Siehe `row().index()`.
- col - Der Spaltenindex der abgefragten Zelle. Siehe `column().index()`.
- settings - Das `DataTables.Settings`-Objekt für die betroffene Tabelle. Dieses kann verwendet werden um eine API-Instanz zu erhalten, falls nötig.

Rückgabewert:

Der Rückgabewert wird nicht verwendet, wenn 'set' als type gewählt wurde. Ansonsten wird der Rückgabewert für die abgefragten Daten verwendet.

### 3.2.158 Beispiele

Lese Tabellendaten aus Objekten:

```
// JSON-Struktur für jede Zeile in diesem Beispiel:
// {
//   "engine": {value},
//   "browser": {value},
//   "platform": {value},
//   "version": {value},
//   "grade": {value}
// }
$('#example').dataTable( {
  "ajaxSource": "sources/objects.txt",
  "columns": [
    { "data": "engine" },
    { "data": "browser" },
    { "data": "platform" },
    { "data": "version" },
    { "data": "grade" }
  ]
} );
```

Lese Informationen aus verschachtelten Objekten:



```
// JSON-Struktur für jede Zeile:
//  {
//    "engine": {value},
//    "browser": {value},
//    "platform": {
//      "inner": {value}
//    },
//    "details": [
//      {value}, {value}
//    ]
//  }
$('#example').dataTable( {
  "ajaxSource": "sources/deep.txt",
  "columns": [
    { "data": "engine" },
    { "data": "browser" },
    { "data": "platform.inner" },
    { "data": "details.0" },
    { "data": "details.1" }
  ]
} );
```

Lese eine Tabelle aus einer DOM-Quelle in Datenobjekte:

```
$(document).ready(function() {
  $('#example').DataTable({
    "columns": [
      { "data": "name" },
      { "data": "position" },
      { "data": "office" },
      { "data": "age" },
      { "data": "start_date" },
      { "data": "salary" }
    ]
  });
} );
```

Verwende data als Funktion, um verschiedene Informationen für Sortierung, Filterung und Anzeige bereitzustellen. In diesem Fall Währung (Preis):

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": function ( row, type, val, meta ) {
      if (type === 'set') {
        row.price = val;
        // Store the computed display and filter values for efficiency
        row.price_display = val=="" ? "" : "$"+numberFormat(val);
        row.price_filter = val=="" ? "" : "$"+numberFormat(val)+" "+val;
        return;
      }
      else if (type === 'display') {
        return row.price_display;
      }
      else if (type === 'filter') {
        return row.price_filter;
      }
      // 'sort', 'type' and undefined all just use the integer
      return row.price;
    }
  } ]
} );

```

Verwendung des Standardinhalts:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": [ 0 ],
    "data": null,
    "defaultContent": "Click to edit"
  } ]
} );

```

Verwendung von Array-Notation - Ausgeben eines Arrays als Liste:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": [ 0 ],
    "data": "name[ , ]"
  } ]
} );

```

### 3.2.159 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.render

## table.columns.i[].defaultContent

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Oftmals soll eine Spalte statischen Inhalt haben, zum Beispiel einfache Knöpfe zum Editieren und / oder Löschen, mit zugewiesenen Events. Diese Option ist für solche Fälle gedacht - statische Inhalte für Spalten erstellen. Soll dynamischer Inhalt erstellt werden (also basierend auf anderen Daten in der Zeile), so sollte die Option `columns.render` verwendet werden.

Weiterhin kann diese Option beim Laden von JSON-Daten nützlich sein, da der hier gesetzte Wert verwendet wird, wenn der Wert im JSON für die Zelle null ist (dann kann hier zum Beispiel der Wert "Nicht verfügbar" definiert werden).

### 3.2.160 Beispiele

Zeige eine Informationsnachricht für ein Feld, das den Wert null oder undefined haben kann:

```

$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    {
      "data": "first_name", // kann null oder undefined sein
      "defaultContent": "<i>Not set</i>"
    }
  ]
} );

```

Zeige einen leeren String, wenn ein Feld null oder undefined ist:

```

$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    {
      "data": "office", // can be null or undefined
      "defaultContent": ""
    }
  ]
} );

```

Erstelle einen Editierungsknopf in der letzten Spalte:

```

$('#example').dataTable( {
  "columns": [
    null,
    null,
    null,
    {
      "data": null,
      "defaultContent": "<button>Edit</button>"
    }
  ]
} );

```

### 3.2.161 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.data
- columns.render

table.columns.i[].name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bei der Arbeit mit der API von DataTables will man oft einzelne Spalten adressieren können, um mit ihnen zu arbeiten (zum Beispiel um den numerischen Inhalt einer Spalte aufzusummieren). DataTables hat zwei grundlegende Arten, Spalten zu adressieren:

- Über den Spaltenindex (automatisch bei der Initialisierung der Tabelle zugewiesen)
- Über einen Namen (mit dieser Option zugewiesen!)

Dies kann das Arbeiten mit Spalten in der API sehr einfach machen. Soll zum Beispiel auf die Daten der Spalte namens location zugegriffen werden, würde man `table.column( 'location:name' ).data()` benutzen. Das Anhängen des Strings `":name"` zeigt DataTables, dass eine Auswahl über den Spaltennamen vorgenommen werden soll. Siehe die Dokumentation der API-Funktion `column()` für mehr Informationen über Spaltenselektoren.

### 3.2.162 Beispiel

Spaltennamen mit `columns` setzen:

```
$('#example').dataTable( {
  "columns": [
    { "name": "engine" },
    { "name": "browser" },
    { "name": "platform" },
    { "name": "version" },
    { "name": "grade" }
  ]
} );
```

### 3.2.163 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `column()`
- `columns()`
- `column().index()`
- `columns().indexes()`

`table.columns.i[].orderable`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Mit diesem Parameter kann dem Endnutzer die Fähigkeit entzogen werden, eine Spalte zu sortieren. Dies kann für Spalten mit generiertem Inhalt nützlich sein, zum Beispiel wenn die Tabelle Knöpfe zum Löschen oder Editieren enthält.

Man beachte, dass diese Option nur die Sortierfähigkeit des Endnutzers beeinträchtigt. Entwickler können die Spalte weiterhin über die Option `order` oder die API-Methode `order()` sortieren.

### 3.2.164 Beispiel

Sortieren der ersten Spalte verbieten:

```
$('#example').dataTable( {
  "columns": [
    { "orderable": false },
    null,
    null,
    null,
    null
  ]
} );
```

### 3.2.165 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `order()`
- `column().order()`
- `columns().order()`

Options

- `ordering`
- `order`

`table.columns.i[].orderData`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Erlaubt der Sortierung einer Spalte die Daten, nach denen sortiert wird, entweder aus einer anderen (oftmals verborgenen) Spalte zu holen, oder aus mehreren Spalten.

Ein häufiges Beispiel ist eine Tabelle, welche Spalten für Vor- und Nachnamen nebeneinander enthält. Es ist intuitiv, dass über diese eine Mehrspaltige Sortierung angewendet werden würde. Ein weiteres Beispiel, mit einer einzelnen Zeile, ist der Fall, dass die dem Nutzer angezeigten Daten selber nicht sortierbar sind (eine Spalte mit Bildern), es aber sortierbare Metadaten gibt (z.B. Dateinamen). Man beachte, dass für diesen Zweck alternativ orthogonale Daten verwendet werden können.

Der Wert muss ein Spaltenindex oder ein Array aus Spaltenindices sein.

### 3.2.166 Beispiel

Benutze `orderData` um eine Sortierung über mehrere Spalten mit `columns` zu definieren:

```
$('#example').dataTable( {
  "columns": [
    { "orderData": [ 0, 1 ] },
    { "orderData": 0, },
    { "orderData": [ 2, 3, 4 ] },
    null,
    null
  ]
} );
```

### 3.2.167 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `order()`
- `column().order()`
- `columns().order()`

#### Options

- `ordering`
- `order`

#### `table.columns.i[].orderDataType`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Die primäre Sortiermethode von DataTables (die Option `ordering`) verwendet aus Performanzgründen Daten, die im Speichercache liegen, anstatt die Daten bei jeder Anfrage direkt aus dem DOM zu lesen (was von Natur aus langsam ist). Es gibt jedoch Fälle, in denen trotz der schlechteren Performanz direkt aus dem DOM gelesen werden soll, zum Beispiel wenn die Tabelle Formularelemente enthält, deren Werte durch den Nutzer verändert werden können. Diese Konfigurationsoption wird bereitgestellt, um Plugins das Anbieten dieser Fähigkeit zu erlauben.

Man beachte, dass keine `columns.orderDataType` Plugins in DataTables eingebaut sind. Sie müssen separat hinzugefügt werden. Siehe die Seite der Sortierplugins für DataTables für weitere Informationen.

### 3.2.168 Beispiel

Setze Typ der Sortierungsdaten mit `columns`:

```
$('#example').dataTable( {
  "columns": [
    null,
    null,
    { "orderDataType": "dom-text" },
    { "orderDataType": "dom-text", "type": "numeric" },
    { "orderDataType": "dom-select" },
    { "orderDataType": "dom-checkbox" }
  ]
} );
```

### 3.2.169 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- `ordering`
- `columns.type`

`table.columns.i[.].orderSequence`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [ 'asc', 'desc' ] ]



Die standardmäßige Sortierrichtung sowie das Verhalten des Sortierleiters (z.B. nur aufsteigendes Sortieren erlauben) kann mit diesem Parameter kontrolliert werden.

### 3.2.170 Beispiel

Definiere die angewandte Sortierrichtung mit `orderSequence`:

```
$('#example').dataTable( {
  "columns": [
    null,
    { "orderSequence": [ "asc" ] },
    { "orderSequence": [ "desc", "asc", "asc" ] },
    { "orderSequence": [ "desc" ] },
    null
  ]
} );
```

### 3.2.171 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### API

- `order()`
- `column().order()`
- `columns().order()`

#### Options

- `ordering`
- `order`

#### Erlaubte Werte

- `asc`
- `desc`

#### `table.columns.i[].render`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Eigenschaft modifiziert die Daten, die von DataTables für verschiedene Operationen benutzt werden, wenn sie aus der Datenquelle gelesen werden. `columns.render` kann als nur lesende Partneroption zu `columns.data` betrachtet werden, wobei `columns.data` lesen und schreiben kann (und daher komplexer ist). Wie `columns.data` kann diese Option auf verschiedene Arten gegeben werden, um ihr Verhalten wie unten beschrieben zu beeinflussen.

Die von `columns.render` zurückgegebenen Daten (egal ob die Option als Funktion, Integer oder String verwendet wird) werden von DataTables für den verlangten Datentyp verwendet (man nennt sie auch die *aufgelösten Daten*). Die Fähigkeit von DataTables, verschiedene Daten für seine verschiedenen Operationen zu verlangen, wird bezeichnet als "orthogonale Daten", und ermöglicht die Verwendung von verschiedenen Formen der gleichen Daten für verschiedene Operationen (zum Beispiel kann ein Datum formatiert sein, um es dem Endnutzer anzuzeigen, aber intern als Integer zur Sortierung benutzt werden).

Es gibt drei spezielle Werte, zu denen `columns.render` aufgelöst werden kann:

- Undefined - Der Wert von `columns.defaultContent` wird verwendet. Ist dieser nicht spezifiziert, so wird ein Fehler gegeben.
- null - Der Wert von `columns.defaultContent` wird verwendet. Ist dieser nicht definiert, so wird für angezeigte Daten ein leerer String verwendet. Für alle anderen Datentypen wird null verwendet.
- Funktion - Die Funktion wird ausgeführt und der Rückgabewert verwendet. Seit DataTables 1.10.1 wird die Funktion im gleichen Scope wie das Datenobjekt der Zeile ausgeführt. Dies führt dazu, dass eine Objektinstanz als Datenquelle einer Zeile verwendet werden kann. Siehe die unteren Beispiele für die Verwendung einer Instanz mit DataTables.

## Array zugriff

Wenn diese Option als String verwendet wird (siehe unten), kann man durch sie sowohl mehrere Datenpunkte, als auch einen einzelnen spezifischen Wert aus einem Array erhalten. Es ist wichtig den Unterschied zwischen den beiden Formen zu verstehen, wenn auf Arraydaten im Datenobjekt der Zeile zugegriffen werden soll.

- Einen oder mehr Werte aus einem Array erhalten - Verwende []-Notation
- Ein einzelnes Element aus dem Array erhalten - Verwende Punktnotation - z.B. ".0" um auf das erste Element des Arrays zuzugreifen.

Man betrachte zum Beispiel die folgende Datenstruktur:

```
"access": [
  { "id": "1", "name": "Printer" },
  { "id": "3", "name": "Desktop" },
  { "id": "4", "name": "VMs" }
]
```

Um die name-Eigenschaft der Objekte im Array anzuzeigen, in einer einzelnen Zelle, verwende `"access[,].name"` - dadurch werden die Namen durch Kommas getrennt aneinandergehängt. Das Ergebnis wäre hier also "Printer, Desktop, VMs".

Um auf ein einzelnes Attribut in einem Array zuzugreifen, verwende `".{index}"`. Soll aus dem obigen JSON nur der Name des ersten Elements gelesen werden, würde man also `"access.0.name"` verwenden. Das Ergebnis wäre hier "Printer".

### 3.2.172 Typen

#### integer

Wird als Array-Index der Datenquelle behandelt. So wird die Option standardmäßig von DataTables verwendet (für jede Spalte inkrementiert).

#### string

Lese ein Objekt aus der Datenquelle. Es gibt drei "spezielle" Optionen, die im String verwendet werden können, um zu verändern, wie DataTables die Daten aus dem Quellobjekt liest:

- `.` - Punktierter Javascript-Notation. So wie ein Punkt in Javascript verwendet wird, um aus geschachtelten Objekten zu lesen, kann er auch in data verwendet werden. Zum Beispiel: `browser.version` oder `browser.name`. Wenn der Name des Objektparameters einen Punkt enthält, kann er mit `\\` maskiert werden. Zum Beispiel `first\\.name`
- `[]` - Array Notation. DataTables kann automatisch Daten aus einer Array-Quelle zusammenführen, wobei die Daten mit dem zwischen den eckigen Klammern angegebenen String verknüpft werden. Zum Beispiel: `name[, ]` würde aus dem Quellarray eine mit Kommas getrennte Liste machen. Wird zwischen den Klammern kein String angegeben, so wird das originale Quellarray zurückgegeben.
- `()` - Funktionsnotation. An das Ende eines Parameters `()` anzufügen führt eine Funktion des gegebenen Namens aus. Zum Beispiel: `browser()` für eine simple Funktion auf der Datenquelle, `browser.version()` für eine Funktion in einer geschachtelten Eigenschaft, oder sogar `browser().version` für die Eigenschaft eines von der Funktion zurückgegebenen Objekts. Man beachte, dass die Funktionsnotation für die Verwendung in `columns.render` statt in `columns.data` empfohlen ist, da sie als Renderer deutlich einfacher zu benutzen ist.

#### objekt

Verwende verschiedene Daten für die verschiedenen von DataTables verlangten Datentypen (`filter`, `display`, `type` oder `sort`). Die Namen der Eigenschaften des Objekts stehen für den jeweiligen Datentyp und die Werte können als `integer`, `string` oder Funktion nach den gleichen Regeln wie sonst in `columns.render` definiert werden.

Man beachte, dass eine Optionale `"_"`-Eigenschaft spezifiziert werden kann. Diese definiert den Standardwert, der verwendet wird, wenn für den von DataTables verlangten Datentyp keine Eigenschaft definiert ist. Ist für den verlangten Datentyp keine Eigenschaft definiert und die `"_"`-Eigenschaft wurde nicht spezifiziert, so wird der Datenwert verwendet, auf den `columns.data` zeigt.

Ein mögliches Beispiel:

```
"data": "phone",
"render": {
  "_": "plain",
  "filter": "filter",
  "display": "display"
}
```

#### funktion `render( data, type, row, meta )`

Die gegebene Funktion wird aufgerufen wann immer DataTables den Wert einer Zelle der Spalte holen muss. Man beachte, dass diese Funktion mehrmals aufgerufen werden kann, da DataTables sie für jeden benötigten Datentyp aufruft - `sort`, `filter` und `display`.

Parameter:

1. Name: data, Typ: Beliebig, Optional: Nein  
Die Daten der Zelle (basierend auf columns.data).
2. Name: type, Typ: String, Optional: Nein  
Der verlangte Datentyp. Dieses Attribut wird für die Unterstützung von orthogonalen Daten in DataTables verwendet. Der Wert ist einer der folgenden:
  - "filter": Hole den Wert, den DataTables für die Filterung auf dieser Zelle verwenden soll.
  - "display": Der Wert, der in der Tabelle angezeigt werden soll.
  - "type": Der Wert, der für Typerkennung verwendet werden soll. Dieser sollte normalerweise mit dem Wert von sort übereinstimmen - siehe die Dokumentation des Typerkennungs-Plugins.
  - undefiniert: Verwende die originalen (unmodifizierten) Daten für die Zelle.
  - Eigener Wert: Es ist möglich für Plugins wie Responsive (durch die Option responsive.orthogonal) oder Buttons (buttons.exportData()), vom Entwickler definierte Datentypen zu verlangen. Dies kann nützlich sein, wenn bestimmte Daten an eine spezifische Erweiterung geschickt werden sollen.

Siehe auch die Methode cell().render(), welche verwendet werden kann, um die gegebene Rendermethode zu beliebigen Zeitpunkten auszuführen.

3. Name: row, Typ: Beliebig, Optional: Nein  
Die komplette Datenquelle der Zeile (nicht basierend auf columns.data).
4. Name: meta, Typ: Objekt, Optional: Nein  
**Seit 1.10.1:** Ein Objekt, welches zusätzliche Informationen über die abgefragte Zelle enthält. Dieses Objekt hat die folgenden Eigenschaften:
  - row - Der Zeilenindex der abgefragten Zelle. Siehe row().index().
  - col - Der Spaltenindex der abgefragten Zelle. Siehe column().index().
  - settings - Das DataTables.Settings-Objekt für die betroffene Tabelle. Dieses kann verwendet werden um eine API-Instanz zu erhalten, falls nötig.

Rückgabewert:

Der Rückgabewert der Funktion wird für die verlangten Daten verwendet.

### 3.2.173 Beispiele

Eine durch Kommas getrennte Liste aus einem Array aus Objekten erstellen:

```
$('#example').dataTable( {  
  "ajaxSource": "sources/deep.txt",  
  "columns": [  
    { "data": "engine" },  
    { "data": "browser" },  
    {  
      "data": "platform",  
      "render": "[, ].name"  
    }  
  ]  
});
```

Als Objekt, extrahiere verschiedene Daten für die verschiedenen Typen:

```
// Dies würde benutzt werden mit einer Datenquelle der Form:
//   { "phone": 5552368, "phone_filter": "5552368 555-2368", "phone_display":
//     "555-2368", ... }
// Hier wird der `phone`-Integer für Sortierung und Typerkennung verwendet, während
// `phone_filter`
// (welches beide Formen enthält) für die Filterung verwendet wird, wenn der Nutzer
// eine der beiden Formen eingibt, während
// die formatierte Telefonnummer in der Tabelle angezeigt wird.
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": null, // Nutze das ganze Quellobjekt als Quelle des Renderers
    "render": {
      "_": "phone",
      "filter": "phone_filter",
      "display": "phone_display"
    }
  } ]
} );
```

Wie oben, mit der 'phone'-Information als Objekt:

```
// Dies würde benutzt werden mit einer Datenquelle der Form:
//   "phone": { "plain": 5552368, "filter": "5552368 555-2368", "display":
//     "555-2368" }
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": 'phone',
    "render": {
      "_": "plain",
      "filter": "filter",
      "display": "display"
    }
  } ]
} );
```

Verwende eine Funktion um einen Link aus der Datenquelle zu konstruieren:

```
$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 0,
    "data": "download_link",
    "render": function ( data, type, row, meta ) {
      return '<a href="' + data + '">Download</a>';
    }
  } ]
} );
```

Zeige Punkte für lange Strings:

```

$('#example').dataTable( {
  "columnDefs": [ {
    "targets": 4,
    "data": "description",
    "render": function ( data, type, row, meta ) {
      return type === 'display' && data.length > 40 ?
        '<span title="'+data+'"'>'+data.substr( 0, 38 )+'...</span>' :
        data;
    }
  } ]
} );

```

Eine Objektinstanz als Datenquelle der Zeile verwenden:

```

function Person( name, age, position ) {
  this._name    = name;
  this._age     = age;
  this._position = position;

  this.name = function () {
    return this._name;
  };

  this.age = function () {
    return this._age;
  };

  this.position = function () {
    return this._position;
  };
}

$(document).ready(function() {
  var table = $('#example').DataTable({
    columns: [
      { data: null, render: 'name' },
      { data: null, render: 'age' },
      { data: null, render: 'position' }
    ]
  });

  table.row.add( new Person( 'Airi Satou',    33, 'Accountant' ) );
  table.row.add( new Person( 'Angelica Ramos', 47, 'Chief Executive Officer
(CEO)' ) );
  table.row.add( new Person( 'Ashton Cox',    66, 'Junior Technical Author' ) );
  table.draw();
} );

```

### 3.2.174 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- columns.data

table.columns.i[.searchable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Mit diesem Parameter kann festgelegt werden, ob DataTables diese Spalte in die filtrierbaren Daten der Tabelle miteinbezieht. Es kann zum Beispiel Sinn machen, diese Option für generierte Spalten, wie Knöpfe zum Editieren und Löschen, zu deaktivieren.

### 3.2.175 Beispiel

Suche auf der ersten Spalte mit columns deaktivieren:

```
$('#example').dataTable( {
  "columns": [
    { "searchable": false },
    null,
    null,
    null,
    null
  ] } );
} );
```

### 3.2.176 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()
- column().search()
- columns().search()

## Options

- search

## table.columns.i[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Wert gelesen aus der Kopfzeile der Spalte. ]

Die Spaltentitel werden normalerweise direkt aus dem DOM gelesen (von den Zellen im THEAD-Element), es kann aber oft nützlich sein, entweder den bestehenden Wert zu überschreiben, oder DataTables tatsächlich einen Header mit Spaltentiteln erstellen zu lassen (zum Beispiel wenn die Tabelle vor der Konstruktion von DataTables kein THEAD-Element hat). Diese Option bietet diese Möglichkeit.

Man beachte, dass DataTables nur einen einfachen Header mit einer einzelnen Zelle pro Spalte erstellen kann. Komplexe Header mit colspan und rowspan Attributen müssen entweder bereits im Dokument definiert sein, oder mit den normalen DOM / jQuery Methoden konstruiert werden.

### 3.2.177 Beispiel

Setze den Titel der ersten Spalte mit columns:

```
$('#example').dataTable( {
  "columns": [
    { "title": "My column title" },
    null,
    null,
    null,
    null
  ]
} );
```

## table.columns.i[].type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bei der Ausführung im Modus klientseitiger Verarbeitung kann DataTables die Daten, die für die Anzeige in den Zellen benutzt werden, auf eine zur auszuführenden Aktion passenden Art verarbeiten. Zum Beispiel werden HTML-Tags aus den Strings entfernt, die für die Filtersuche benutzt werden, während die Sortierung potentiell Währungssymbole entfernt, um Geldwerte numerisch Sortieren zu können. Die ausgeführte Formatierungsaktion, die die Daten *normalisiert*, damit sie für Sortierung und Suche verwendbar werden, hängt vom *Typ* der Spalte ab.

DataTables hat eine Reihe von eingebauten Typen, die automatisch erkannt werden:

- **date:** Datums- / Zeitwerte. Man beachte, dass das von DataTables eingebaute Datumsparsen die Javascript-Methode `Date.parse()` verwendet, welche nur eine begrenzte Menge an Daten unterstützt. Zusätzliche Unterstützung für Datumsformatierung kann durch Plugins hinzugefügt werden.
  - Sortierung - Chronologisch sortiert
  - Filterung - Kein Effekt
- **num:** Simple Zahlensortierung.
  - Sortierung - Numerisch sortiert
  - Filterung - Kein Effekt
- **num-fmt:** Numerisches Sortieren von formatierten Zahlen. Zahlen mit Tausendertrennzeichen, Währungssymbolen oder Prozentzeichen werden von DataTables automatisch numerisch sortiert.
  - Unterstützte eingebaute Währungssymbole sind \$, £, € und ¥.
  - Unterstützte eingebaute Tausendertrennzeichen sind ' und ,
  - Beispiele:
    - \$100,000 - Sortiert als 100000
    - £10'000 - Sortiert als 10000
    - 5'000 - Sortiert als 5000
    - 40% - Sortiert als 40
  - Sortierung - Numerisch sortiert
  - Filterung - Kein Effekt
- **html-num:** Wie num, nur mit zusätzlichen HTML-Tags in den Daten.
  - Sortierung - Numerisch sortiert
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **html-num-fmt:** wie num-fmt, nur mit zusätzlichen HTML-Tags in den Daten.
  - Sortierung - Numerisch sortiert
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **html:** Einfache Verarbeitung von Strings mit HTML-Tags.
  - Sortierung - Sortiert mit entfernten HTML-Tags
  - Filterung - HTML-Tags aus gefiltertem String entfernt
- **string:** Rückfalltyp, falls die Daten der Spalte nicht zu den Anforderungen von einem der obigen Typen passen.
  - Sortierung - Kein Effekt
  - Filterung - Kein Effekt

Es wird erwartet, dass die obigen Optionen die meisten von DataTables verwendeten Datentypen abdecken, da Daten aber flexibel sind und in vielen Formen kommen, können zusätzliche Typen mit anderen Effekten durch Plugins hinzugefügt werden. So kann fast jedes vorstellbare Datenformat sortiert werden!

Als Optimierung kann der Spaltentyp, wenn er schon im voraus bekannt ist, über diesen Parameter gesetzt werden. Das erspart DataTables das Ausführen der Erkennungsroutine.

Es ist zu beachten, dass dieser Parameter bei serverseitiger Bearbeitung (serverSide) keinen Effekt hat, da Sortierung und Filterung auf dem Server durchgeführt werden.

### 3.2.178 Beispiel

Setze manuell den Typ der ersten Spalte mit columns:

```
$('#example').dataTable( {
  "columns": [
    { "type": "html" },
    null,
    null,
    null,
    null
  ]
} );
```

### 3.2.179 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ordering
- searching

table.columns.i[].visible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

DataTable kann durch diese Option und die API-Methoden `column().visible()` / `columns().visible()` Spalten dynamisch anzeigen und verbergen. Diese Option kann verwendet werden, um den initialen Zustand der Sichtbarkeit der Spalte zu setzen, während die API-Methoden den Zustand zu einem späteren Zeitpunkt verändern können.

Dies ist besonders nützlich wenn die Tabelle eine große Anzahl Spalten hat, da so dem Nutzer die Möglichkeit gegeben werden kann, selber zu entscheiden, welche Spalten angezeigt werden sollen. Alternativ können mit dieser Option Spalten verborgen werden, die der Nutzer nicht sehen soll (zum Beispiel eine Spalte mit IDs einer Datenbank).

### 3.2.180 Beispiel

Die erste Spalte mit columns verbergen:

```
$('#example').dataTable( {
  "columns": [
    { "visible": false },
    null,
    null,
    null,
    null
  ] } );
} );
```

### 3.2.181 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `column().visible()`
- `columns().visible()`

Options

- `scrollX`

`table.columns.i[[]].width`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dieser Parameter kann verwendet werden, um die Breite einer Spalte zu definieren, und kann jeden CSS-Wert akzeptieren (3em, 20px, etc.).

Man beachte, dass auf den Pixel genaue Spaltenbreite in Tabellen mit dynamischem Inhalt praktisch unmöglich ist, weshalb es nicht verwunderlich ist, wenn die tatsächliche Breite der Spalte um ein paar Pixel von dem hier definierten Wert abweicht. Die Spaltenbreite in der Tabelle hängt von vielen Aspekten wie den Zellenrändern, den

Tabellenrändern, der border-collapse-Eigenschaft, dem Inhalt und anderem ab. Sowohl DataTables als auch der Browser versuchen unter Betrachtung all dieser Aspekte ein optimales Layout für die Tabelle zu finden.

### 3.2.182 Beispiel

Setze über columns die Breite der ersten Spalte auf 20%:

```
$('#example').dataTable( {
  "columns": [
    { "width": "20%" },
    null,
    null,
    null,
    null
  ]
} );
```

### 3.2.183 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- columns.adjust()

Options

- autoWidth

table.language.aria.paginate.first

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die ARIA-Beschriftung für den 'Erste Seite'-Knopf der eingebauten Paginierungssteuerung. Dies ist besonders nützlich, um Zugänglichkeit zu erhalten, wenn der Knopf selbst ein Icon wie « anzeigen soll.

### 3.2.184 Beispiel

Definiere die Optionen von language.paginate als Icons mit ARIA-Beschriftungen:

```

$('#example').DataTable( {
  pagingType: 'full',
  language: {
    paginate: {
      first:    '«',
      previous: '<',
      next:     '>',
      last:     '»'
    },
    aria: {
      paginate: {
        first:    'First',
        previous: 'Previous',
        next:     'Next',
        last:     'Last'
      }
    }
  }
} );

```

### 3.2.185 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.aria.paginate
- language.paginate.first

table.language.aria.paginate.last

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die ARIA-Beschriftung für den 'Letzte Seite'-Knopf der eingebauten Paginierungssteuerung. Dies ist besonders nützlich, um Zugänglichkeit zu erhalten, wenn der Knopf selbst ein Icon wie » anzeigen soll.

### 3.2.186 Beispiel

Definiere die Optionen von language.paginate als Icons mit ARIA-Beschriftungen:

```

$('#example').DataTable( {
  pagingType: 'full',
  language: {
    paginate: {
      first:    '«',
      previous: '<',
      next:     '>',
      last:     '»'
    },
    aria: {
      paginate: {
        first:    'First',
        previous: 'Previous',
        next:     'Next',
        last:     'Last'
      }
    }
  }
} );

```

### 3.2.187 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.aria.paginate
- language.paginate.last

#### table.language.aria.paginate.next

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die ARIA-Beschriftung für den 'Nächste Seite'-Knopf der eingebauten Paginierungssteuerung. Dies ist besonders nützlich, um Zugänglichkeit zu erhalten, wenn der Knopf selbst ein Icon wie » oder › anzeigen soll.

### 3.2.188 Beispiel

Benutze Icons für die Knöpfe zur nächsten und vorherigen Seite mit ARIA-Beschriftungen:

```

$('#example').DataTable( {
  language: {
    paginate: {
      previous: '<',
      next: '>'
    },
    aria: {
      paginate: {
        previous: 'Previous',
        next: 'Next'
      }
    }
  }
} );

```

### 3.2.189 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.aria.paginate
- language.paginate.next

#### table.language.aria.paginate.previous

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die ARIA-Beschriftung für den 'Vorherige Seite'-Knopf der eingebauten Paginierungssteuerung. Dies ist besonders nützlich, um Zugänglichkeit zu erhalten, wenn der Knopf selbst ein Icon wie « oder < anzeigen soll.

### 3.2.190 Beispiel

Benutze Icons für die knöpfe zur nächsten und vorherigen Seite mit ARIA-Beschriftungen:

```

$('#example').DataTable( {
  language: {
    paginate: {
      previous: '<',
      next: '>'
    },
    aria: {
      paginate: {
        previous: 'Previous',
        next: 'Next'
      }
    }
  }
} );

```

### 3.2.191 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language.aria.paginate
- language.paginate.previous

#### table.language.aria.paginate

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dieses Objekt enthält die Eigenschaften, die für die ARIA-Beschriftungen der eingebauten Paginierungssteuerung von DataTables verwendet werden. Wenn die language.paginate Optionen verwendet werden, um auf den Knöpfen der Paginierungssteuerung Icons statt Text anzuzeigen, kann mit den ARIA-Beschriftungen der Erhalt der Zugänglichkeit gewährleistet werden.

Es ist anzumerken, dass ARIA-Beschriftungen nicht nötig sind, wenn die language.paginate Optionen reinen Text enthalten. Da dies standardmäßig der Fall ist, haben die Optionen von language.aria.paginate standardmäßig keine Werte.

### 3.2.192 Beispiel

Definiere die Optionen von language.paginate als Icons mit ARIA-Beschriftungen:



```

$('#example').DataTable( {
  pagingType: 'full',
  language: {
    paginate: {
      first:    '«',
      previous: '<',
      next:     '>',
      last:     '»'
    },
    aria: {
      paginate: {
        first:    'First',
        previous: 'Previous',
        next:     'Next',
        last:     'Last'
      }
    }
  }
} );

```

### 3.2.193 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### Options

- language
- language.aria.paginate.next
- language.aria.paginate.previous
- language.aria.paginate.first
- language.aria.paginate.last

#### table.language.aria.sortAscending

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ": activate to sort column ascending" ]

ARIA-Beschriftung, die zu den Elementen der Kopfzeile einer Tabelle hinzugefügt wird, wenn die Spalte durch Aktivieren der Spalte aufsteigend sortiert werden kann (durch anklicken oder betätigen der Enter Taste wenn die Spalte im Fokus ist).

Es ist anzumerken, dass der Text im Spaltenkopf als Präfix zu diesem String gesetzt wird.

### 3.2.194 Beispiel

ARIA-String für aufsteigende Sortierung setzen:

```
$('#example').dataTable( {
  "language": {
    "aria": {
      "sortAscending": " - click/return to sort ascending"
    }
  }
});
```

### 3.2.195 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.aria
- language.aria.sortDescending

table.language.aria.sortDescending

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ": activate to sort column descending" ]

ARIA-Beschriftung, die zu den Elementen der Kopfzeile einer Tabelle hinzugefügt wird, wenn die Spalte durch Aktivieren der Spalte absteigend sortiert werden kann (durch anklicken oder betätigen der Enter Taste wenn die Spalte im Fokus ist).

Es ist anzumerken, dass der Text im Spaltenkopf als Präfix zu diesem String gesetzt wird.

### 3.2.196 Beispiel

ARIA-String für absteigende Sortierung setzen:

```
$('#example').dataTable( {
  "language": {
    "aria": {
      "sortDescending": " - click/return to sort descending"
    }
  }
} );
```

### 3.2.197 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.aria
- language.aria.sortAscending

table.language.aria

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Strings die nur für WAI-ARIA Beschriftungen und Steuerungen verwendet werden (diese sind auf der Seite nicht sichtbar, werden aber von Bildschirmlesern vorgelesen und müssen daher auch internationalisiert werden).

### 3.2.198 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.aria.sortAscending
- language.aria.sortDescending

table.language.decimal

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "" ]

In Javascript wird die Dezimalstelle mit einem Punkt (.) markiert. Viele Teile der Welt verwenden aber ein Komma (,) oder andere Zeichen wie den Unicode Dezimalseparator (?) oder einen Bindestrich (-) um die Dezimalstelle in einer angezeigten Zahl zu markieren.

So formatierte Zahlen werden von Javascript nicht automatisch als Zahlen erkannt, aber die Typerkennung und die Sortiermethoden von DataTables können durch die `language.decimal` Option informiert werden, welches Zeichen als Dezimalstelle in Zahlen dient. So können die Typerkennung und Sortieralgorithmen von DataTables korrekt mit diesen Zahlen arbeiten.

Diese Option ist etwas ungewöhnlich, da DataTables nie eine formatierte Fließkommazahl anzeigt (es ist nicht nötig!). Diese Option beeinflusst also nur das Parsen der gelesenen Daten (keine der anderen Sprachoptionen hat diese Fähigkeit).

Jedes Zeichen kann mit dieser Option als Dezimalstelle gesetzt werden, wobei die Dezimalstellen einer einzelnen Tabelle konsistent sein müssen (also können nicht Zahlen mit Punkten und Zahlen mit Kommas als Dezimalstelle in der gleichen Tabelle verwendet werden). Verschiedene Tabellen auf der gleichen Seite können verschiedene Dezimalzeichen verwenden, falls nötig.

Wenn ein leerer String gegeben wird (der Standard für diesen Parameter), wird ein Punkt (.) als Dezimalzeichen angenommen.

### 3.2.199 Beispiel

Erkenne und sortiere Zahlen die ein Komma als Dezimalzeichen verwenden:

```
$('#example').dataTable( {
  "language": {
    "decimal": ",",
  }
});
```

Erkenne und sortiere Zahlen die einen Bindestrich als Dezimalzeichen verwenden:

```
$('#example').dataTable( {
  "language": {
    "decimal": "-",
    "thousands": "."
  }
});
```

### 3.2.200 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.thousands

#### table.language.emptyTable

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "No data available in table" ]

Dieser String wird statt language.zeroRecords angezeigt, wenn die Tabelle unabhängig von der Filterung keine Daten enthält - also wenn es keine Einträge im Datensatz der Tabelle gibt.

Es ist anzumerken, dass dies ein optionaler Parameter ist. Ist er nicht gegeben, so wird stattdessen der Wert von language.zeroRecords verwendet (entweder der gegebene oder der standardmäßige Wert).

### 3.2.201 Beispiel

Setze einen String für emptyTable:

```
$('#example').dataTable( {
  "language": {
    "emptyTable": "No data available in table"
  }
} );
```

### 3.2.202 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.infoEmpty
- language.zeroRecords

## table.language.info

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Showing \_START\_ to \_END\_ of \_TOTAL\_ entries" ]

Dieser String informiert den Endnutzer über die momentan in der Tabelle angezeigten Daten. Die folgenden Tokens können im String verwendet werden, und werden dynamisch ersetzt, wenn die Tabellenanzeige aktualisiert wird.

Diese Tokens können an beliebigen Stellen im String platziert werden, oder entfernt wenn die Sprachbedingungen es verlangen:

- `_START_` - Anzeigindex des ersten Eintrags auf der momentanen Seite
- `_END_` - Anzeigindex des letzten Eintrags auf der momentanen Seite
- `_TOTAL_` - Anzahl der Einträge in der Tabelle nach Filterung
- `_MAX_` - Anzahl der Einträge in der Tabelle ohne Filterung
- `_PAGE_` - Momentane Seitennummer
- `_PAGES_` - Gesamtzahl Seiten in der Tabelle

### 3.2.203 Beispiel

Zeige Seitenzahl:

```
$('#example').dataTable( {
  "language": {
    "info": "Showing page _PAGE_ of _PAGES_"
  }
} );
```

### 3.2.204 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.infoEmpty
- language.infoFiltered
- language.infoPostFix
- language.thousands

## table.language.infoEmpty

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Showing 0 to 0 of 0 entries" ]

Informationsstring der angezeigt werden soll wenn die Tabelle leer ist. Für gewöhnlich sollte das Format dieses Strings zu info passen.

## 3.2.205 Beispiel

Zeige Seitenzahl:

```
$('#example').dataTable( {
  "language": {
    "infoEmpty": "No entries to show"
  }
} );
```

## 3.2.206 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.info
- language.infoFiltered
- language.infoPostFix
- language.thousands

## table.language.infoFiltered

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "(filtered from \_MAX\_ total entries)" ]

Wenn ein Nutzer die Daten in einer Tabelle filtert, wird dieser String an die Information (language.info) angehängt, um eine Vorstellung zu geben, wie stark die Filterung ist.

Das Token `_MAX_` wird dynamisch aktualisiert - siehe `language.info` für Informationen über alle verfügbaren Tokens.

### 3.2.207 Beispiel

Ergebnis der Filterung zusammenfassender Informationsstring:

```
$('#example').dataTable( {
  "language": {
    "infoFiltered": " - filtered from _MAX_ records"
  }
} );
```

### 3.2.208 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.info
- language.infoEmpty
- language.infoPostFix
- language.thousands

#### table.language.infoPostFix

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: "" ]

Es kann manchmal nützlich sein, zusätzliche Informationen an den Informationsstring zu hängen, und diese Variable tut genau das.

Dies String wird immer an language.info (, language.infoEmpty und language.infoFiltered, in welcher Reihenfolge sie auch vorliegen mögen,) angehängt werden.

### 3.2.209 Beispiel

Postfixer String zur Zusammenfassung der Informationen:

```
$('#example').dataTable( {
  "language": {
    "infoPostFix": "All records shown are derived from real information."
  }
} );
```

### 3.2.210 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.info
- language.infoEmpty
- language.infoFiltered
- language.thousands

#### table.language.lengthMenu

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Show \_MENU\_ entries" ]

Beschreibt die Aktion, die durchgeführt wird, wenn sich das Drop-down-Menü der Optionen für die Seitenlänge ändert. Das Token `_MENU_` wird durch die in `lengthMenu` spezifizierte Liste an Auswahlmöglichkeiten ersetzt, und kann durch eine eigene Auswahlliste ersetzt werden, falls nötig.

### 3.2.211 Beispiele

Reine Änderung der Formulierung:

```
$('#example').dataTable( {
  "language": {
    "lengthMenu": "Display _MENU_ records"
  }
});
```

Änderung der Formulierung und der Optionen (hier sollte aber besser lengthMenu verwendet werden!):

```
$('#example').dataTable( {
  "language": {
    "lengthMenu": 'Display <select>'+
      '<option value="10">10</option>'+
      '<option value="20">20</option>'+
      '<option value="30">30</option>'+
      '<option value="40">40</option>'+
      '<option value="50">50</option>'+
      '<option value="-1">All</option>'+
      '</select> records'
  }
});
```

### 3.2.212 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- page.len()

Options

- language
- lengthChange
- lengthMenu

table.language.loadingRecords

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: "Loading..." ]

Bei der Verwendung von Ajax-Datenquellen und während dem ersten Zeichnen, wenn DataTables die Daten sammelt, wird diese Nachricht in einer leeren Zeile der Tabelle angezeigt, um den Nutzer wissen zu lassen, dass die Daten geladen werden. Man beachte, dass dieser Parameter nicht bei der serverseitigen Bearbeitung verwendet wird, sondern nur bei Ajax-Datenquellen und klientseitiger Bearbeitung.

### 3.2.213 Beispiel

Setze Text der Ladeinformation:

```
$('#example').dataTable( {
  "ajax": "json.txt",
  "language": {
    "loadingRecords": "Please wait - loading..."
  }
} );
```

### 3.2.214 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- ajax
- language
- language.zeroRecords

#### table.language.paginate.first

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "First" ]

Verwendeter Text für den Knopf, der den Nutzer zur ersten Seite führt.

Es ist anzumerken, dass DataTables diese Eigenschaft als HTML ins Dokument schreibt, es können also HTML-Tags im String benutzt werden, HTML-Entities wie < oder > sollten aber als jeweils &lt; oder &gt; kodiert werden.

### 3.2.215 Beispiel

Setze Text für den 'Erste Seite'-Knopf:

```
$('#example').dataTable( {
  "language": {
    "paginate": {
      "first": "First page"
    }
  }
} );
```

### 3.2.216 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.paginate
- language.paginate.last
- language.paginate.next
- language.paginate.previous
- language.aria.paginate
- language.aria.paginate.first

table.language.paginate.last

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Last" ]

Verwendeter Text für den Knopf, der den Nutzer zur letzten Seite führt.

Es ist anzumerken, dass DataTables diese Eigenschaft als HTML ins Dokument schreibt, es können also HTML-Tags im String benutzt werden, HTML-Entities wie < oder > sollten aber als jeweils &lt; oder &gt; kodiert werden.

### 3.2.217 Beispiel

Setze Text für den 'Letzte Seite'-Knopf:

```
$('#example').dataTable( {
  "language": {
    "paginate": {
      "last": "Last page"
    }
  }
} );
```

### 3.2.218 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.paginate
- language.paginate.first
- language.paginate.next
- language.paginate.previous
- language.aria.paginate
- language.aria.paginate.last

table.language.paginate.next

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Next" ]

Verwendeter Text für den Knopf, der den Nutzer zur nächsten Seite führt.

Es ist anzumerken, dass DataTables diese Eigenschaft als HTML ins Dokument schreibt, es können also HTML-Tags im String benutzt werden, HTML-Entities wie < oder > sollten aber als jeweils &lt; oder &gt; kodiert werden.

### 3.2.219 Beispiel

Setze Text für den 'Nächste Seite'-Knopf:

```

$('#example').dataTable( {
  "language": {
    "paginate": {
      "next": "Next page"
    }
  }
} );

```

### 3.2.220 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.paginate
- language.paginate.first
- language.paginate.last
- language.paginate.previous
- language.aria.paginate
- language.aria.paginate.next

table.language.paginate.previous

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Previous" ]

Verwendeter Text für den Knopf, der den Nutzer zur vorherigen Seite führt.

Es ist anzumerken, dass DataTables diese Eigenschaft als HTML ins Dokument schreibt, es können also HTML-Tags im String benutzt werden, HTML-Entities wie < oder > sollten aber als jeweils &lt; oder &gt; kodiert werden.

### 3.2.221 Beispiel

Setze Text für den 'Vorherige Seite'-Knopf:

```
$('#example').dataTable( {
  "language": {
    "paginate": {
      "previous": "Previous page"
    }
  }
});
```

### 3.2.222 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### Options

- language
- language.paginate
- language.paginate.first
- language.paginate.next
- language.paginate.last
- language.aria.paginate
- language.aria.paginate.previous

#### table.language.paginate

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Strings die von DataTables für die eingebauten Typen der Paginierungssteuerung verwendet werden.

### 3.2.223 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

#### Options

- language
- language.paginate.first
- language.paginate.last
- language.paginate.next
- language.paginate.previous

## table.language.processing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Processing..." ]

Text, der angezeigt wird, während die Tabelle eine Nutzeraktion bearbeitet (normalerweise ein Sortierungskommando oder etwas ähnliches).

## 3.2.224 Beispiel

Setze Text der Bearbeitungsinformation:

```
$('#example').dataTable( {
  "language": {
    "processing": "DataTables is currently busy"
  }
} );
```

## 3.2.225 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.loadingRecords

## table.language.search

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



## Schlagworte / Tags

[ Default-Wert: "Search:" ]

Setzt den String, der von DataTables als Beschriftung des Eingabefelds für den Filter angezeigt wird.

Das Token `_INPUT_`, falls es im String benutzt wird, wird durch die HTML-Textbox des Eingabefelds ersetzt, wodurch kontrolliert werden kann, wo es in Relation zur Beschriftung erscheint. Wird `_INPUT_` nicht verwendet, so wird das Eingabefeld automatisch an den String angehängt.

### 3.2.226 Beispiele

Eingabefeld wird automatisch angehängt:

```
$('#example').dataTable( {
  "language": {
    "search": "Filter records:"
  }
} );
```

Spezifiziere wo das Eingabefeld erscheinen soll:

```
$('#example').dataTable( {
  "language": {
    "search": "Apply filter _INPUT_ to table"
  }
} );
```

### 3.2.227 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `search()`

Options

- `language`
- `searching`
- `language.searchPlaceholder`

`table.language.searchPlaceholder`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "" ]

HTML5 hat ein placeholder-Attribut für Elemente mit input type="text" eingeführt, um zusätzliche Informationen für Eingabefelder zu zeigen, wenn diese keinen Wert haben.

Dieser Parameter kann verwendet werden, um einen Wert für das placeholder-Attribut des Eingabefelds der Suche von DataTable zu setzen.

### 3.2.228 Beispiele

Setze einen Platzhalter:

```
$('#example').dataTable( {
  language: {
    searchPlaceholder: "Search records"
  }
} );
```

Eingabefeld ohne Beschriftung - nur mit Platzhalter:

```
$('#example').dataTable( {
  language: {
    search: "_INPUT_",
    searchPlaceholder: "Search..."
  }
} );
```

### 3.2.229 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()

Options

- language
- searching
- language.search

table.language.thousands

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ", " ]

Der eingebaute Zahlenformatierer von DataTables (formatNumber) wird verwendet, um große Zahlen in der Tabelleninformation zu formatieren. Standardmäßig wird ein Komma verwendet, durch diesen Parameter kann aber trivial zu einem beliebigen anderen Zeichen gewechselt werden um zu einer Lokalisierung zu passen. Alternativ kann ein leerer String angegeben werden, falls keine Tausendertrennzeichen erwünscht sind.

Es ist anzumerken, dass das Tausendertrennzeichen - anders als das Kommazeichen von language.decimal - nur zur Ausgabe von Informationen verwendet wird (in der info Option, um genau zu sein). Es zu verändern ändert nicht wie DataTables numerische Werte liest.

### 3.2.230 Beispiel

Setze ein einfaches Anführungszeichen als Tausendertrennzeichen:

```
$('#example').dataTable( {
  "language": {
    "thousands": ""
  }
} );
```

### 3.2.231 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- language.decimal
- language.info
- language.infoEmpty
- language.infoFiltered
- language.infoPostFix
- formatNumber

table.language.url

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Alle von DataTables bereitgestellten Sprachoptionen können in einer Datei auf dem Server gespeichert werden, welche DataTables abrufen wenn dieser Parameter übergeben wird. Die Datei muss eine gültige JSON-Datei sein, und das darin enthaltene Objekt muss die gleichen Eigenschaften wie das language-Objekt im Initialisierungsobjekt haben.

Eine breite Auswahl an entsprechenden Übersetzungen ist auf der Webseite des DataTables-Frameworks verfügbar.

Es ist anzumerken, dass, wenn dieser Parameter gesetzt wurde, die Initialisierung von DataTables aufgrund der Ajax-Datenlast asynchron sein wird. Die Tabelle wird also nicht gezeichnet, bis das Ajax-Request abgeschlossen ist. Aktionen, die erst ausgeführt werden können, wenn die Tabelle mit ihrer Initialisierung fertig ist, sollten daher in den initComplete-Callback platziert werden.

### 3.2.232 Beispiele

Lade Sprachinformationen aus einer Remote-Datei:

```
$('#example').dataTable( {
  "language": {
    "url": "/dataTables/i18n/de_de.lang"
  }
} );
```

Lade Sprachinformationen aus einer Remote.Datei, verwende initComplete-Callback:

```
$('#example').dataTable( {
  "language": {
    "url": "/dataTables/i18n/de_de.lang"
  },
  "initComplete": function () {
    var api = this.api();

    // Put the sum of column 5 into the footer cell
    $( api.column( 5 ).footer() ).html(
      api.column( 5 ).data().reduce( function (a, b) {
        return a + b;
      } )
    );
  }
} );
```

### 3.2.233 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- language
- initComplete

table.language.zeroRecords

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "No matching records found" ]

Text, der in den Tabelleinträgen angezeigt wird, wenn nach der Filterung keine Informationen mehr anzuzeigen sind.

Es ist anzumerken, dass language.emptyTable angezeigt wird, wenn die Tabelle gar keine Informationen enthält (ungeachtet der momentanen Filterung), während dieser Parameter verwendet wird, wenn die Tabelle aufgrund der Filterung leer ist.

### 3.2.234 Beispiel

Setze String für durch Filterung geleerte Tabelle:

```
$('#example').dataTable( {
  "language": {
    "zeroRecords": "No records to display"
  }
} );
```

### 3.2.235 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- search()

Options

- language
- language.emptyTable
- search

### table.\$.fn.dataTable.ext.errMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn DataTables auf einen Fehler trifft den es "kennt" (zum Beispiel ungültiges JSON), so kann es nützliche Informationen zum Fehler und seiner Behebung ausgeben.

Die standardmäßige Handhabung von Fehlern ist absichtlich aufdringlich um Entwickler während dem Schreiben von mit DataTables interagierendem Code wissen zu lassen, dass es Probleme gibt. Da dies jedoch potentiell nicht das gewünschte Verhalten auf der Seite ist, kann die Fehlerbehandlung über diese Option kontrolliert werden.

## 3.2.236 Typen

### string

Führe eine der eingebauten Aktionen der Fehlerbehandlung aus:

- alert (standard) - Den Fehler über alert ausgeben
- throw - Einen Javascript-Fehler werfen
- none - Nichts tun (In diesem Fall sollte das error-Event verwendet werden)

### Funktion \$.fn.dataTable.ext.errMode( settings, techNote, message )

Als Funktion kann eine eigene Fehlerbehandlung definiert werden, zum Beispiel mit Anzeigen einer Nachricht zum Endnutzer oder neu laden der Seite um sich von einem Fehler zu erholen.

Parameter:

1. Name: settings, Typ: DataTables.Settings, Optional: Nein  
DataTables Einstellungsobjekt
2. Name: techNote, Typ: integer, Optional: Nein  
Referenz zur technischen Notiz welche detaillierte Informationen zum aufgetretenen Fehler enthält, zu finden unter <https://datatables.net/tn/{techNote}>
3. Name: message, Typ: string, Optional: Nein  
Fehlernachricht

Rückgabewert:

Keine Rückgabe wird erwartet oder verwendet.

### 3.2.237 Beispiel

DataTables Fehler werfen lassen statt sie über alert() auszugeben:

```
$.fn.dataTable.ext.errMode = 'throw';

$('#example').dataTable( {
  ajax: 'data.json'
} );
```

### 3.2.238 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Events

- error

Erlaubte Werte

- alert
- throw
- none

table.colReorder.fixedColumnsLeft

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die ColReorder-Erweiterung für DataTables voraus.

Wenn das Ändern der Reihenfolge von Spalten in der Tabelle erlaubt ist, sollen oftmals bestimmte Spalten in ihrer Position fest sein (zum Beispiel zum Festsetzen einer Index- Such- oder Aktionsspalte am Anfang der Tabelle). Diese Option ermöglicht es, Spalten angefangen vom linken Rand der Tabelle festzusetzen (colReorder.fixedColumnsRight gibt die gleiche Möglichkeit von rechts).

Dies ist besonders nützlich, wenn ColReorder zusammen mit der FixedColumns-Erweiterung verwendet wird.

Der Wert dieser Option gibt an, wieviele Spalten, angefangen vom linken Rand der Tabelle, in ihrer Position unveränderbar sein sollen.

### 3.2.239 Beispiel

Verhindere das Ändern der Position der ersten Spalte:

```
$('#example').DataTable( {
  colReorder: {
    fixedColumnsLeft: 1
  }
} );
```

### 3.2.240 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- colReorder.fixedColumnsRight

table.colReorder.fixedColumnsRight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die ColReorder-Erweiterung für DataTables voraus.

Wenn das Ändern der Reihenfolge von Spalten in der Tabelle erlaubt ist, sollen oftmals bestimmte Spalten in ihrer Position fest sein (zum Beispiel zum Festsetzen einer Index- Such- oder Aktionsspalte am Anfang der Tabelle). Diese Option ermöglicht es, Spalten angefangen vom rechten Rand der Tabelle festzusetzen (colReorder.fixedColumnsLeft gibt die gleiche Möglichkeit von links).

Dies ist besonders nützlich, wenn ColReorder zusammen mit der FixedColumns-Erweiterung verwendet wird.

Der Wert dieser Option gibt an, wieviele Spalten, angefangen vom rechten Rand der Tabelle, in ihrer Position unveränderbar sein sollen.

### 3.2.241 Beispiel

Verhindere das Ändern der Positionen der letzten beiden Spalten:



```
$('#example').DataTable( {
  colReorder: {
    fixedColumnsRight: 2
  }
} );
```

### 3.2.242 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

Options

- colReorder.fixedColumnsLeft

#### table.colReorder.order

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Man beachte: Dieses Attribut setzt die ColReorder-Erweiterung für DataTables voraus.

Diese Option gibt die Möglichkeit, eine standardmäßige Reihenfolge der Spalten einer Tabelle zu definieren. Normalerweise sollen die Spalten in der Reihenfolge bleiben, die im HTML oder im gespeicherten Zustand (stateSave) definiert ist, aber falls nötig kann über diese Option eine initiale Reihenfolge definiert werden.

### 3.2.243 Typ

array

Ein Array aus Integer-Werten welches die Reihenfolge der Spalten definiert. Die Position im Array ist die Position die eine Spalte einnimmt, und der Wert ist der momentane Spaltenindex der Spalte, die an dieser Position angezeigt werden soll.

Das Array *muss* alle Spalten der Tabelle platzieren und darf keine Duplikate enthalten.

### 3.2.244 Beispiel

ColReorder aktivieren und die Reihenfolge der Spalten umkehren (in einer Tabelle mit sechs Spalten):

```
$('#example').DataTable( {
  colReorder: {
    order: [ 5, 4, 3, 2, 1, 0 ]
  }
} );
```

### 3.2.245 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- colReorder.order()

#### table.colReorder.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Man beachte: Dieses Attribut setzt die ColReorder-Erweiterung für DataTables voraus.

ColReorder gibt dem Endnutzer (standardmäßig) visuelles Feedback über die Änderung der Reihenfolge indem eine Einfügungsmarkierung angezeigt wird und die anderen Spalten während des Ziehens ihre Position ändern. Durch diese Option kann das letztere Feedback deaktiviert werden. Dies kann für ältere Browser oder Computer bei komplexen Tabellen sinnvoll sein, da es die Performanz beeinträchtigen könnte.

### 3.2.246 Beispiel

Deaktiviere visuelles Feedback der Spaltenpositionen in Echtzeit:

```
$('#example').DataTable( {
  colReorder: {
    realtime: false
  }
} );
```

## table.colReorder

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

ColReorder gibt Endnutzern die Möglichkeit, die Reihenfolge der Spalten in der Tabelle durch Ziehen mit der Maus zu verändern. Entwicklern gibt ColReorder die Möglichkeit, das gleiche durch die API zu tun.

Diese Option ermöglicht es, ColReorder für DataTables zu aktivieren und zu konfigurieren. Einfach als Boolean-Wert `true` gegeben aktiviert diese Option ColReorder mit den Standardeinstellungen (die in `$.fn.dataTable.ColReorder.defaults` definiert sind). Die Option kann auch als Objekt gegeben werden, um ColReorder mit eigenen Konfigurationen zu starten.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von ColReorder, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.2.247 Typen

#### boolean

Als boolean gegeben kann dieser Parameter ColReorder für die Tabelle aktivieren, die gerade erstellt wird. `true` aktiviert colReorder, `false` aktiviert colReorder nicht.

Dies ist eine Abkürzung um ColReorder mit der Standardkonfiguration zu aktivieren. Veränderungen an der Konfiguration können vorgenommen werden, indem der Parameter als Objekt übergeben wird. Siehe unten.

#### objekt

Als Objekt gegeben wird ColReorder für die Tabelle aktiviert, wobei die Standardwerte (`$.fn.dataTable.ColReorder.defaults`) erweitert und potentiell von den in diesem Objekt gegebenen Optionen überschrieben werden. So kann colReorder für jede Tabelle einzeln konfiguriert werden.

### 3.2.248 Beispiele

ColReorder für eine Tabelle aktivieren:

```
$('#example').DataTable( {
  colReorder: true
} );
```

ColReorder mit eigenen Konfigurationen aktivieren:

```
$('#example').DataTable( {
  colReorder: {
    realtime: false
  }
} );
```

### table.fixedColumns.heightMatch

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'semiauto' ]

Man beachte: Dieses Attribut setzt die FixedColumns-Erweiterung für DataTables voraus.

FixedColumns geht so vor, dass es die Zellen der originalen Tabelle kloniert und sie visuell über der Tabelle positioniert in das Dokument einfügt - wodurch es aussieht, als ob sie fixiert wären.

Da die fixierten Spalten von der Haupttabelle separate Tabellen sind, müssen die Zeilenhöhen der beiden angeglichen werden - sonst kann es zu Fehlausrichtungen kommen.

FixedColumns hat hierfür drei Optionen zur Auswahl:

- none - DataTables nimmt keine Höhenangleichung vor (in diesem Fall kann die Angleichung stattdessen durch CSS erzwungen werden).
- semiauto - Die Höhenangleichung wird einmal durchgeführt und das Ergebnis für wiederholte Verwendung im Cache gespeichert.
- auto - Die Höhenangleichung wird bei jedem Draw durchgeführt (am langsamsten aber genauesten).

### 3.2.249 Beispiel

Erlaube CSS die Zeilenhöhe zu setzen:

```
$('#example').DataTable( {
  fixedColumns: {
    heightMatch: 'none'
  }
} );
```

#### Erlaubte Werte

- none

- semiauto
- auto

### table.fixedColumns.leftColumns

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Man beachte: Dieses Attribut setzt die FixedColumns-Erweiterung für DataTables voraus.

FixedColumns fixiert standardmäßig die Spalte ganz links in der Tabelle, es ist aber auch möglich zwei oder mehr Spalten zu fixieren, je nachdem wie die Anwendung es benötigt. Die Anzahl der zu fixierenden Spalten vom linken Rand aus kann durch diesen Parameter definiert werden.

### 3.2.250 Beispiel

Zwei Spalten am linken Rand fixieren:

```
$('#example').DataTable( {
  fixedColumns: {
    leftColumns: 2
  }
} );
```

Keine Spalten am linken Rand fixieren:

```
$('#example').DataTable( {
  fixedColumns: {
    leftColumns: 0,
    rightColumns: 1
  }
} );
```

### table.fixedColumns.rightColumns

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die FixedColumns-Erweiterung für DataTables voraus.

FixedColumns kann Spalten auf der linken und der rechten Seite der Tabelle fixieren. Standardmäßig wird nur eine Spalte auf der linken Seite fixiert, aber über diesen Parameter kann spezifiziert werden, wie viele Spalten auf der rechten Seite fixiert werden sollen. Der Wert ist ein einfacher Integer, der die Anzahl der Spalten angibt.

### 3.2.251 Beispiele

Spalten am linken und rechten Rand fixiert:

```
$('#example').DataTable( {
  fixedColumns: {
    rightColumns: 1
  }
} );
```

Keine Spalten auf der linken Seite fixiert:

```
$('#example').DataTable( {
  fixedColumns: {
    leftColumns: 0,
    rightColumns: 1
  }
} );
```

#### table.fixedColumns

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Man beachte: Dieses Attribut setzt die FixedColumns-Erweiterung für DataTables voraus.

FixedColumns gibt die Möglichkeit, eine oder mehrere Spalten auf der linken oder rechten Seite einer horizontal scrollenden Tabelle zu fixieren. So bleiben die Informationen in den fixierten Tabellen sichtbar, auch wenn durch den Datensatz gescrollt wird. Dies kann besonders nützlich sein, wenn eine große Anzahl Spalten angezeigt werden soll.

Diese Option ermöglicht es, FixedColumns für DataTables zu aktivieren und zu konfigurieren. Einfach als Boolean-Wert `true` gegeben aktiviert diese Option FixedColumns mit den Standardeinstellungen (die in `$.fn.dataTable.FixedColumns.defaults` definiert sind). Die Option kann auch als Objekt gegeben werden, um FixedColumns mit eigenen Konfigurationen zu starten.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von FixedColumns, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.2.252 Typen

#### boolean

Als boolean gegeben kann dieser Parameter FixedColumns für die Tabelle aktivieren, die gerade erstellt wird. `true` aktiviert FixedColumns, `false` aktiviert FixedColumns nicht.

Dies ist eine Abkürzung um FixedColumns mit der Standardkonfiguration zu aktivieren. Veränderungen an der Konfiguration können vorgenommen werden, indem der Parameter als Objekt übergeben wird. Siehe unten.

#### objekt

Als Objekt gegeben wird FixedColumns für die Tabelle aktiviert, wobei die Standardwerte (`$.fn.dataTable.FixedColumns.defaults`) erweitert und potentiell von den in diesem Objekt gegebenen Optionen überschrieben werden. So kann FixedColumns für jede Tabelle einzeln konfiguriert werden.

### 3.2.253 Beispiele

FixedColumns für eine Tabelle aktivieren:

```
$('#example').DataTable( {  
  fixedColumns: true  
} );
```

FixedColumns mit eigenen Konfigurationen aktivieren:

```
$('#example').DataTable( {  
  fixedColumns: {  
    leftColumns: 2  
  }  
} );
```

#### table.fixedHeader.footer

Datentyp	BOOLEAN

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Man beachte: Dieses Attribut setzt die FixedHeader-Erweiterung für DataTables voraus.

Dieser Parameter gibt die Möglichkeit, die fixierte Fußzeile der FixedHeader-Erweiterung zu aktivieren oder deaktivieren. Ist sie aktiviert, so ist die Fußzeile *klebend*. Sie ist also immer sichtbar während ein Nutzer durch die Tabelle scrollt.

### 3.2.254 Beispiel

Deaktiviere die Kopfzeile und aktiviere die Fußzeile:

```
$('#example').DataTable( {
  fixedHeader: {
    header: false,
    footer: true
  }
} );
```

#### table.fixedHeader.footerOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die FixedHeader-Erweiterung für DataTables voraus.

Obwohl eine fixierte Kopfzeile häufiger benötigt wird, kann eine fixierte Fußzeile doch nützlich sein, um ein Element fixer Position am unteren Ende einer Seite zur Verfügung zu stellen. Dieses Element kann zum Beispiel häufig verwendete Funktionen und Links für den Endnutzer enthalten. Der durch dieses fixe Element eingeführte Offset muss bei der Verwendung von FixedHeader bedacht werden, um das Element an der richtigen Stelle zu platzieren.



Dieser Parameter gibt die Möglichkeit FixedHeader zu informieren, dass ein Offset auf die Fußzeile angewandt werden soll, und wie groß dieser Offset sein soll. Der Wert wird in Pixeln angegeben.

### 3.2.255 Beispiele

Setze den Offset der Fußzeile auf einen bekannten Wert:

```
$('#example').DataTable( {
  fixedHeader: {
    footer: true,
    footerOffset: 40
  }
});
```

Verwende jQuery um einen benötigten Offset zu berechnen:

```
$('#example').DataTable( {
  fixedHeader: {
    footer: true,
    footerOffset: $('#navMenu').outerHeight()
  }
});
```

### 3.2.256 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- fixedHeader.footerOffset()

Options

- fixedHeader.headerOffset

table.fixedHeader.header

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Man beachte: Dieses Attribut setzt die FixedHeader-Erweiterung für DataTables voraus.

Dieser Parameter gibt die Möglichkeit, die fixierte Kopfzeile der FixedHeader-Erweiterung zu aktivieren oder deaktivieren. Ist sie aktiviert, so ist die Kopfzeile *klebend*. Sie ist also immer sichtbar während ein Nutzer durch die Tabelle scrollt.

### 3.2.257 Beispiel

Deaktiviere die Kopfzeile und aktiviere die Fußzeile:

```
$('#example').DataTable( {
  fixedHeader: {
    header: false,
    footer: true
  }
});
```

#### table.fixedHeader.headerOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die FixedHeader-Erweiterung für DataTables voraus.

Viele Webseiten verwenden ein Navigationselement, welches eine fixe Position am oberen Ende der Seite hat, um dem Endnutzer schnellen Zugriff auf häufig verwendete Funktionen und Links zu geben. Dies muss bei der Verwendung von FixedHeader bedacht werden, damit der durch das Element verursachte Offset nicht die korrekte positionierung der fixen Kopfzeile stört.

Dieser Parameter gibt die Möglichkeit FixedHeader zu informieren, dass ein Offset auf die Kopfzeile angewandt werden soll, und wie groß dieser Offset sein soll. Der Wert wird in Pixeln angegeben.

### 3.2.258 Beispiele

Setze den Offset der Kopfzeile auf einen bekannten Wert:

```
$('#example').DataTable( {
  fixedHeader: {
    headerOffset: 50
  }
} );
```

Verwende jQuery um einen benötigten Offset zu berechnen:

```
$('#example').DataTable( {
  fixedHeader: {
    headerOffset: $('#navMenu').outerHeight()
  }
} );
```

### 3.2.259 Verwandt

Die folgenden Optionen stehen in direktem Zusammenhang mit dieser und könnten bei der Entwicklung auch nützlich sein.

API

- `fixedHeader.headerOffset()`

Options

- `fixedHeader.footerOffset`

table.fixedHeader

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Man beachte: Dieses Attribut setzt die FixedHeader-Erweiterung für DataTables voraus.

FixedHeader ist eine Erweiterung für DataTables, welche es ermöglicht, eine Kopf- oder Fußzeile anzuzeigen, die *kleben bleibt*. Das Element fixiert sich also zum oberen oder unteren Ende des Fensters während der Nutzer durch die Tabelle scrollt. Das Ergebnis ist, dass der Nutzer die Kopfzeile / Fußzeile durchgehend sieht, während er durch die Tabelle scrollt. Die fixierten Elemente scrollen dann von der Seite runter, wenn der Nutzer an der Tabelle vorbei scrollt.

Dieser Parameter bietet Konfigurationsoptionen für FixedHeader und ermöglicht simple Initialisation sowie feinere Kontrolle über die individuellen Features von FixedHeader.

### 3.2.260 Typen

#### boolean

Aktiviere FixedHeader mit den standardmäßigen Konfigurationsoptionen (definiert in `$.fn.dataTable.FixedHeader.defaults`).

#### objekt

Feinkörnige Kontrolle über die Initialisierung von FixedHeader. Ermöglicht volle Konfiguration der verfügbaren Optionen. Siehe die Dokumentation der einzelnen Optionen für Details.

### 3.2.261 Beispiele

Aktiviere FixedHeader mit den Standardwerten:

```
$('#example').DataTable( {
  fixedHeader: true
} );
```

Aktiviere FixedHeader mit zusätzlich aktivierter Fußzeile:

```
$('#example').DataTable( {
  fixedHeader: {
    footer: true
  }
} );
```

#### table.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	style, table

Setzt die CSS Style-Klassen an der Tabelle.

Beispiel:

```
dashlet.addOption("table.style", "table-striped");
```

## table.language.locale

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt die Locale der Table.

Beispiel:

```
dashlet.addOption("table.language.locale", "DE");
```

## table.columnDef.i[].className

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option fügt einfach eine Klasse zu jeder Zelle einer Spalte hinzu, ungeachtet ob die Tabellenquelle DOM, Javascript oder Ajax ist. Dies kann für das styling von Spalten nützlich sein.

## 3.2.262 Beispiel

Mit className über columnDefs eine Klasse zu den Zellen der ersten Spalte hinzufügen:

```
$('#example').dataTable( {
  "columnDefs": [
    { className: "my_class", "targets": [ 0 ] }
  ]
} );
```

Dashlet Option

```
dashlet.addOption("table.columns.i[2].className", "none");
```

### table.columnDef.n[].className

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option fügt einfach eine Klasse zu jeder Zelle einer Spalte hinzu, ungeachtet ob die Tabellenquelle DOM, Javascript oder Ajax ist. Dies kann für das styling von Spalten nützlich sein.

### 3.2.263 Beispiel

Mit `className` über `columnDefs` eine Klasse zu den Zellen der ersten Spalte hinzufügen:

```
$('#example').dataTable( {
  "columnDefs": [
    { className: "my_class", "targets": [ 0 ] }
  ]
} );
```

#### Dashlet Option

```
dashlet.addOption("table.columns.n[PRODUCTDESCRIPTION].className", "none");
```

### table.colReorder.enable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Man beachte: Dieses Attribut setzt die ColReorder-Erweiterung für DataTables voraus.

Abhängig vom Status Ihrer Anwendung kann es hilfreich sein, die Benutzereingabesteuerung von ColReorder zu bestimmten Zeiten zu deaktivieren. Diese Option bietet diese Möglichkeit beim erstmaligen Erstellen der Tabelle, während die Methoden `colReorder.enable ()` und `colReorder.disable ()` die Option zum Aktivieren der Benutzerinteraktion nach dem Erstellen der Tabelle bieten.

### 3.2.264 Beispiel

Deaktivieren ColReorder bei der Initialisierung:

```
$('#example').DataTable( {
  colReorder: {
    enable: false
  }
} );
```

#### table.complex.header

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.3
<b>Schlagworte / Tags</b>	

Zeigt einen komplexen Header für die Datentabelle mit Gruppierung an.

### 3.2.265 Beispiel

```
dashlet.addOption("table.complex.header", "<th colspan="3"><div style="margin-bottom: 10px; height: 40px; background-color: red;"><b>Cars Trains Ships and more</b></div></th"
+ "><th colspan="5"><div style="margin-bottom: 10px; height: 40px; background-color: yellow;"><b>Classic Models</b></div></th>");
```

#### table.ext.search

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Aktivieren Sie die benutzerdefinierte Suche in DataTable.

Beispiel:

```
<span style="font-size:10.0pt"><span style="color:black">dashlet.addOption(</span></span><span style="font-size:10.0pt"><span style="color:#2a00ff">"table.ext.search"</span></span></span><span style="font-size:10.0pt"><span style="color:black">,</span></span></span><span style="font-size:10.0pt"><span style="color:#2a00ff">"true"</span></span></span><span style="font-size:10.0pt"><span style="color:black">);</span></span></span>
```

Siehe auch:

table.ext.search.{x}  
table.ext.search.startSearchMode  
table.ext.search.searchModelList  
table.ext.search.searchModeButtonNames  
table.ext.search.searchModeButtonVisible  
table.ext.search.caseinsensitiveSearch  
table.ext.search.caseinsensitiveCheckBoxVisible  
table.ext.search.caseinsensitiveText  
table.ext.search.inputBarShow  
table.ext.search.inputBarMax  
table.ext.search.inputBarMaxtext  
table.ext.search.inputBarMin  
table.ext.search.inputBarMintext  
table.ext.search.minmaxColumnIndex  
table.ext.search.minmaxColumnType  
table.ext.search.minmaxColumnName  
table.ext.search.hiddenSearch

table.ext.search.minmaxColumnIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>



<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definieren die SpaltenIndex für die Min/Max Suche in dataTable

Beispiel:

```
dashlet.addOption("table.ext.search.minmaxColumnIndex", "2");
```

Siehe auch:

table.ext.search  
 table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

table.ext.search.startSearchMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: smart ]

Definieren einen Startmodus für die benutzerdefinierte Suche in dataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.startSearchMode", "exact");
```

Siehe auch:

table.ext.search  
 table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

Erlaubte Werte

- exact
- like
- smart

table.ext.search.searchModeList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: ['like','exact','smart']]

Definiert die Liste der benutzerdefinierten Suchoptionen in DataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.searchModeList",
  "['like','exact','smart','custom1','custom2']");
```

Siehe auch:

table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.searchModeButtonNames

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definiert die Namensliste für die benutzerdefinierten Suchoptionen in DataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.searchModeButtonNames", "[ 'ähnliche Suche', 'exakte Suche', 'Smarte Suche', 'Eigene Suche Frankfurt', 'Eigene Suche USA' ]");
```

Siehe auch:

table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax

table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.searchModeButtonVisible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px; font-weight: bold;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Stellt den Button für die benutzerdefinierte Suche als sichtbar oder unsichtbar ein.

Beispiel:

```
<span style="font-size:10.0pt"><span style="color:black">dashlet.addOption(</span><span style="color:#2a00ff">"</span></span><span style="font-size:10.0pt"><span style="color:#2a00ff">"</span><span style="color:black">,</span><span style="color:#2a00ff">"true"</span></span><span style="font-size:10.0pt"><span style="color:black">);</span></span>
```

Siehe auch:

table.ext.search  
 table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex

table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.caseinsensitiveSearch

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #2e8b57; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Aktivieren Sie die Groß-/Kleinschreibungssuche in DataTable.

Beispiel:

```
<span style="font-size:10.0pt"><span style="color:black">dashlet.addOption(</span><span style="color:#2a00ff">"</span></span><span style="font-size:10.0pt"><span style="color:#2a00ff">"</span><span style="color:black">,</span><span style="color:#2a00ff">"true"</span><span style="color:black">);</span></span>
```

Siehe auch:

table.ext.search

table.ext.search.{x}

table.ext.search.startSearchMode

table.ext.search.searchModeList

table.ext.search.searchModeButtonNames

table.ext.search.searchModeButtonVisible

table.ext.search.caseinsensitiveSearch

table.ext.search.caseinsensitiveCheckBoxVisible

table.ext.search.caseinsensitiveText

table.ext.search.inputBarShow

table.ext.search.inputBarMax

table.ext.search.inputBarMaxtext

table.ext.search.inputBarMin

table.ext.search.inputBarMintext

table.ext.search.minmaxColumnIndex

table.ext.search.minmaxColumnType

table.ext.search.minmaxColumnName

table.ext.search.hiddenSearch

## table.ext.search.caseinsensitiveCheckBoxVisible

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Stellt den CheckBox für die Groß/Kienschreibung wechsel in der benutzerdefinierte Suche als sichtbar oder unsichtbar ein.

Beispiel:

```
dashlet.addOption("table.ext.search.caseinsensitiveCheckBoxVisible", "true");
```

Siehe auch:

table.ext.search

table.ext.search.{x}

table.ext.search.startSearchMode

table.ext.search.searchModeList

table.ext.search.searchModeButtonNames

table.ext.search.searchModeButtonVisible

table.ext.search.caseinsensitiveSearch

table.ext.search.caseinsensitiveCheckBoxVisible

table.ext.search.caseinsensitiveText

table.ext.search.inputBarShow

table.ext.search.inputBarMax

table.ext.search.inputBarMaxtext

table.ext.search.inputBarMin

table.ext.search.inputBarMintext

table.ext.search.minmaxColumnIndex

table.ext.search.minmaxColumnType

table.ext.search.minmaxColumnName

table.ext.search.hiddenSearch

## table.ext.search.caseinsensitiveText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definiert den benutzerdefinierten Namen für die caseSensitive-Suche in DataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.caseinsensitiveText", "Groß- Klein Egal");
```

Siehe auch:

table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.inputBarShow

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Aktiviert die Leiste, um die maximale/minimale benutzerdefinierte Suche in dataTable zu definieren.

Beispiel:

```
dashlet.addOption("table.ext.search.inputBarShow", "true");
```

Siehe auch:

[table.ext.search.{x}](#)  
[table.ext.search.startSearchMode](#)  
[table.ext.search.searchModeList](#)  
[table.ext.search.searchModeButtonNames](#)  
[table.ext.search.searchModeButtonVisible](#)  
[table.ext.search.caseinsensitiveSearch](#)  
[table.ext.search.caseinsensitiveCheckBoxVisible](#)  
[table.ext.search.caseinsensitiveText](#)  
[table.ext.search.inputBarShow](#)  
[table.ext.search.inputBarMax](#)  
[table.ext.search.inputBarMaxtext](#)  
[table.ext.search.inputBarMin](#)  
[table.ext.search.inputBarMintext](#)  
[table.ext.search.minmaxColumnIndex](#)  
[table.ext.search.minmaxColumnType](#)  
[table.ext.search.minmaxColumnName](#)  
[table.ext.search.hiddenSearch](#)

### table.ext.search.inputBarMax

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Aktivieret die Leiste, um die maximale benutzerdefinierte Suche in dataTable zu definieren.

Beispiel:

```
dashlet.addOption("table.ext.search.inputBarMax", "true");
```

Siehe auch:

[table.ext.search.{x}](#)  
[table.ext.search.startSearchMode](#)  
[table.ext.search.searchModeList](#)  
[table.ext.search.searchModeButtonNames](#)  
[table.ext.search.searchModeButtonVisible](#)  
[table.ext.search.caseinsensitiveSearch](#)  
[table.ext.search.caseinsensitiveCheckBoxVisible](#)  
[table.ext.search.caseinsensitiveText](#)  
[table.ext.search.inputBarShow](#)



table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.inputBarMin

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

[default: false ]

Aktiviert die Leiste, um die minimale benutzerdefinierte Suche in dataTable zu definieren.

Beispiel:

```
dashlet.addOption("table.ext.search.inputBarMin", "true");
```

Siehe auch:

table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

## table.ext.search.inputBarMaxtext

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definiert den benutzerdefinierten Namen für die Maximale-Suche in DataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.inputBarMaxtext", "Maximum:");
```

Siehe auch:

[table.ext.search.{x}](#)  
[table.ext.search.startSearchMode](#)  
[table.ext.search.searchModeList](#)  
[table.ext.search.searchModeButtonNames](#)  
[table.ext.search.searchModeButtonVisible](#)  
[table.ext.search.caseinsensitiveSearch](#)  
[table.ext.search.caseinsensitiveCheckBoxVisible](#)  
[table.ext.search.caseinsensitiveText](#)  
[table.ext.search.inputBarShow](#)  
[table.ext.search.inputBarMax](#)  
[table.ext.search.inputBarMaxtext](#)  
[table.ext.search.inputBarMin](#)  
[table.ext.search.inputBarMintext](#)  
[table.ext.search.minmaxColumnIndex](#)  
[table.ext.search.minmaxColumnType](#)  
[table.ext.search.minmaxColumnName](#)  
[table.ext.search.hiddenSearch](#)

## table.ext.search.inputBarMintext

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.23

### Schlagworte / Tags

Definiert den benutzerdefinierten Namen für die Minimale-Suche in DataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.inputBarMintext", "Maximum:");
```

Siehe auch:

[table.ext.search.{x}](#)  
[table.ext.search.startSearchMode](#)  
[table.ext.search.searchModeList](#)  
[table.ext.search.searchModeButtonNames](#)  
[table.ext.search.searchModeButtonVisible](#)  
[table.ext.search.caseinsensitiveSearch](#)  
[table.ext.search.caseinsensitiveCheckBoxVisible](#)  
[table.ext.search.caseinsensitiveText](#)  
[table.ext.search.inputBarShow](#)  
[table.ext.search.inputBarMax](#)  
[table.ext.search.inputBarMaxtext](#)  
[table.ext.search.inputBarMin](#)  
[table.ext.search.inputBarMintext](#)  
[table.ext.search.minmaxColumnIndex](#)  
[table.ext.search.minmaxColumnType](#)  
[table.ext.search.minmaxColumnName](#)  
[table.ext.search.hiddenSearch](#)

### table.ext.search.minmaxColumnName

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definieren die SpaltenName für die Min/Max Suche in dataTable

Beispiel:

```
dashlet.addOption("table.ext.search.minmaxColumnName", "Customer Name");
```

Siehe auch:

table.ext.search  
 table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.minmaxColumnType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Definieren die DatenTyp des Spaltens für die Min/Max Suche in dataTable

Beispiel:

```
dashlet.addOption("table.ext.search.minmaxColumnType", "other");
```

Siehe auch:

table.ext.search  
 table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax

table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

### table.ext.search.hiddenSearch

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.23
<b>Schlagworte / Tags</b>	

Sichtbarkeit der benutzerdefinierten Suche in dataTable.

Beispiel:

```
dashlet.addOption("table.ext.search.hiddenSearch", "true");
```

Siehe auch:

table.ext.search.{x}  
 table.ext.search.startSearchMode  
 table.ext.search.searchModeList  
 table.ext.search.searchModeButtonNames  
 table.ext.search.searchModeButtonVisible  
 table.ext.search.caseinsensitiveSearch  
 table.ext.search.caseinsensitiveCheckBoxVisible  
 table.ext.search.caseinsensitiveText  
 table.ext.search.inputBarShow  
 table.ext.search.inputBarMax  
 table.ext.search.inputBarMaxtext  
 table.ext.search.inputBarMin  
 table.ext.search.inputBarMintext  
 table.ext.search.minmaxColumnIndex  
 table.ext.search.minmaxColumnType  
 table.ext.search.minmaxColumnName  
 table.ext.search.hiddenSearch

## 3.3 DataTables Responsive

### 3.3.1 table.columns.i[].responsivePriority

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: undefiniert. Priorität der Spalte wird von Responsive automatisch auf 10000 gestellt. ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Responsive entfernt automatisch Spalten von der rechten Seite der Tabelle wenn die Tabelle zu breit für ein gegebenes Display ist. Dies ist zwar eine nützliche Standardeinstellung, aber vielleicht sollen die Spalten der Tabelle in einer eigens definierten Reihenfolge verborgen werden. Dieser Parameter gibt diese Möglichkeit, indem er den Spalten eine Priorität der Sichtbarkeit zuweisen kann.

Der gegebene Wert setzt die Priorität der Sichtbarkeit der Spalte, wobei eine niedrigere Zahl für eine höhere Priorität beim sichtbar bleiben steht. Zum Beispiel wird eine Spalte mit Priorität 2 vor einer Spalte mit Priorität 1 entfernt, egal in welcher Reihenfolge sie in der Tabelle stehen.

Dies kann besonders nützlich sein, wenn die Spalte(n) auf der rechten Seite sichtbar bleiben sollen, weil sie Knöpfe oder andere wichtige Informationen enthalten.

Die Spaltenpriorität kann auch durch das Attribut `data-priority` in der Kopfzelle der Spalte definiert werden (zum Beispiel `<th data-priority="1">First name</th>`). Sind sowohl das `data-priority` Attribut als auch der Wert von `columns.responsivePriority` für eine Spalte gesetzt, so wird immer der Wert von `columns.responsivePriority` verwendet.

Es wird erwartet, dass die gegebenen Werte größer gleich 0 sind, aber negative Werte sind technisch gesehen möglich, um die Priorität wichtiger Spalten zu erhöhen.

### 3.3.2 Beispiele

Setze die Priorität der ersten und letzten Spalte:

```
$('#example').DataTable( {
  responsive: true,
  columnDefs: [
    { responsivePriority: 1, targets: 0 },
    { responsivePriority: 2, targets: -1 }
  ]
} );
```

Verberge Spalten von links nach rechts:

```

$('#example').DataTable( {
  responsive: true,
  columns: [
    { responsivePriority: 6 },
    { responsivePriority: 5 },
    { responsivePriority: 4 },
    { responsivePriority: 3 },
    { responsivePriority: 2 },
    { responsivePriority: 1 }
  ]
} );

```

### table.columnDef.t[].responsivePriority

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: undefiniert. Priorität der Spalte wird von Responsive automatisch auf 10000 gestellt. ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Responsive entfernt automatisch Spalten von der rechten Seite der Tabelle wenn die Tabelle zu breit für ein gegebenes Display ist. Dies ist zwar eine nützliche Standardeinstellung, aber vielleicht sollen die Spalten der Tabelle in einer eigens definierten Reihenfolge verborgen werden. Dieser Parameter gibt diese Möglichkeit, indem er den Spalten eine Priorität der Sichtbarkeit zuweisen kann.

Der gegebene Wert setzt die Priorität der Sichtbarkeit der Spalte, wobei eine niedrigere Zahl für eine höhere Priorität beim sichtbar bleiben steht. Zum Beispiel wird eine Spalte mit Priorität 2 vor einer Spalte mit Priorität 1 entfernt, egal in welcher Reihenfolge sie in der Tabelle stehen.

Dies kann besonders nützlich sein, wenn die Spalte(n) auf der rechten Seite sichtbar bleiben sollen, weil sie Knöpfe oder andere wichtige Informationen enthalten.

Die Spaltenpriorität kann auch durch das Attribut data-priority in der Kopfzelle der Spalte definiert werden (zum Beispiel `<th data-priority="1">First name</th>`). Sind sowohl das data-priority Attribut als auch der Wert von columns.responsivePriority für eine Spalte gesetzt, so wird immer der Wert von columns.responsivePriority verwendet.

Es wird erwartet, dass die gegebenen Werte größer gleich 0 sind, aber negative Werte sind technisch gesehen möglich, um die Priorität wichtiger Spalten zu erhöhen.

### 3.3.3 Beispiele

Setze die Priorität der ersten und letzten Spalte:

```

$('#example').DataTable( {
  responsive: true,
  columnDefs: [
    { responsivePriority: 1, targets: 0 },
    { responsivePriority: 2, targets: -1 }
  ]
} );

```

Verberge Spalten von links nach rechts:

```

$('#example').DataTable( {
  responsive: true,
  columns: [
    { responsivePriority: 6 },
    { responsivePriority: 5 },
    { responsivePriority: 4 },
    { responsivePriority: 3 },
    { responsivePriority: 2 },
    { responsivePriority: 1 }
  ]
} );

```

### table.responsive.breakpoints

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.fn.dataTable.Responsive.breakpoints ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Die Sichtbarkeit von Spalten in DataTables mit aktiviertem Responsive kann über Breakpoints und zu diesen Breakpoints passenden Klassennamen (und anderen logischen Operationen) kontrolliert werden. So kann genau kontrolliert werden, welche Spalten einer Tabelle für welchen Gerätetyp sichtbar sind. Siehe die Dokumentation von Responsive für weitere Informationen.

Die von einer Instanz von Responsive verwendeten Breakpoints sind einzigartig für diese Instanz (also jede Tabelle), wodurch für verschiedene Tabellen verschiedene Breakpoints gesetzt werden können, falls dies nötig ist.

Die standardmäßigen Breakpoints aus \$.fn.dataTable.Responsive.breakpoints sind:



```
[
  { name: 'desktop', width: Infinity },
  { name: 'tablet-l', width: 1024 },
  { name: 'tablet-p', width: 768 },
  { name: 'mobile-l', width: 480 },
  { name: 'mobile-p', width: 320 }
]
```

Wird ein eigenes Array an Breakpoints definiert, so ist die Reihenfolge der Breakpoints unwichtig. Responsive bringt das Array vor der Verwendung automatisch in die intern benötigte Reihenfolge.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.3.4 Typ

#### array

Ein Array aus Objekten, in dem jedes Objekt zwei Eigenschaften hat, welche den Breakpoint beschreiben:

- **name** - Der Name des Breakpoints für Bezug in Klassen. Kann ein einzelnes Wort sein, oder mit Anhängen von `-[a-z]` ein Sub-Breakpoint, ähnlich wie in den Standardwerten, um zum Beispiel Breakpoints für `tablet`, `tablet landscape` und `tablet portrait` zu definieren
- **width** - Die Breite (in Pixeln) des Viewports der Seite, bei der dieser Breakpoint verwendet wird. Der angegebene Wert ist die maximale Größe, für die dieser Breakpoint verwendet wird, und er wird verwendet, bis der nächste Breakpoint gefunden wird. Bei den obigen Standardwerten wird zum Beispiel der `tablet-l` Breakpoint für  $768 < x \leq 1024$  verwendet (wobei  $x$  die Breite des Viewports ist).

### 3.3.5 Beispiele

Setze eigene Breakpoints bei der Initialisierung von DataTables:

```
$('#example').DataTable( {
  responsive: {
    breakpoints: [
      { name: 'desktop', width: Infinity },
      { name: 'tablet', width: 1024 },
      { name: 'fablet', width: 768 },
      { name: 'phone', width: 480 }
    ]
  }
} );
```

Setze eigene Breakpoints über die Standardwerte:

```
$.fn.dataTable.Responsive.breakpoints = [
  { name: 'desktop', width: Infinity },
  { name: 'tablet', width: 1024 },
  { name: 'fablet', width: 768 },
  { name: 'phone', width: 480 }
];

$('#example').DataTable( {
  responsive: true
} );
```

### table.responsive.details

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Responsive kann das Kinderzeilen-Feature von DataTables benutzen um Informationen über aus dem Display entfernte Spalten als Kinderzeilen anzuzeigen. Dies kann besonders nützlich sein, um komplexe Informationen auf Geräten mit kleinem Bildschirm darzustellen. Siehe die Dokumentation von Responsive für weitere Informationen.

Diese Option ermöglicht es, zu konfigurieren, wie Responsive mit den Detailzeilen umgeht. Als Boolean kann das Anzeigen der Detailzeile mit dieser Option aktiviert oder deaktiviert werden, als Objekt kann kontrolliert werden, wie die Zeilen angezeigt werden und ob die Kindzeilen angezeigt oder verborgen werden sollen.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.3.6 Typen

#### boolean

Als Boolean aktiviert und deaktiviert diese Option die Anzeige der Kindzeilen von Responsive. false deaktiviert die Kindzeilen komplett (Spalten werden einfach entfernt und ihr Inhalt ist nur noch über die API erreichbar), während true die Kindzeilen mit den Standardwerten aus dem Objekt `$.fn.dataTables.Responsive.defaults.details` aktiviert.

## object

Als Objekt gegeben wird die Anzeige der Kindzeilen von Responsive aktiviert und die im Objekt gegebenen Einstellungen erweitern oder überschreiben die Standardeinstellungen.

### 3.3.7 Beispiele

Deaktiviere die Anzeige der Detailzeilen:

```
$('#example').DataTable( {
  responsive: {
    details: false
  }
});
```

Setze die Option type für die Anzeige:

```
$('#example').DataTable( {
  responsive: {
    details: {
      type: 'column'
    }
  }
});
```

#### table.responsive.details.display

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.fn.dataTable.Responsive.display.childRow ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Responsive kann Informationen über verborgene Spalten in den Kindzeilen von DataTables anzeigen (`row().child()`), aber vielleicht sollen die Daten auf eine andere Art angezeigt werden (potentiell damit die Kindzeilen für andere Funktionen wie Editierung verfügbar sind) - dieser Parameter gibt diese Möglichkeit.

Die gegebene Funktion ist dafür verantwortlich, die Daten anzuzeigen und zu verbergen, normalerweise wenn der Endnutzer dies verlangt (zweiter Parameter). Sie kann verwendet werden, um die Daten in Kindzeilen anzuzeigen (wie im Standard), in einem modalen pop-up, in einem separaten Informationselement oder sogar potentiell in einem neuen Fenster mit einer Detailansicht.

Responsive hat eine Reihe von eingebauten Anzeigefunktionen die durch das Objekt `$.fn.dataTable.Responsive.display` erreicht werden können - die Optionen sind:

- `childRow` - Zeige verborgene Informationen in einer Kindzeile, deren Sichtbarkeit vom Endnutzer bestimmt werden kann.
- `childRowImmediate` - Zeige die Informationen in einer Kindzeile, aber warte nicht auf die Anfrage des Nutzers um die Daten zu zeigen, zeige sie einfach direkt.
- `modal()` - Zeige Informationen in einem modalen pop-up - es ist zu bemerken, dass dies eine **Funktion** ist und bei der Zuweisung ausgeführt werden sollte (siehe Beispiel unten). So können Optionen in die verwendete Modal-Library übergeben werden. Bei der Verwendung mit den Bootstrap, Foundation und JQueryUI Integrationsoptionen benutzt Responsive das native modale Display des Frameworks um sicherzustellen, dass dem Endnutzer ein konsistentes Interface angezeigt wird. Die Optionen, die in die Funktion übergeben werden können, hängen vom verwendeten Framework ab:
  - DataTables, Bootstrap und Foundation:
    - `header` - Eine Funktion, die einen String zurückgeben sollte, der in der Kopfzeile des Modals angezeigt werden soll. Ein einzelner Parameter wird übergeben: Die `row()`-Instanz der Zeile, deren Details angezeigt werden.
  - jQuery UI:
    - `header` - Eine Funktion, die einen String zurückgeben sollte, der in der Kopfzeile des Modals angezeigt werden soll. Ein einzelner Parameter wird übergeben: Die `row()`-Instanz der Zeile, deren Details angezeigt werden.
    - `dialog` - Ein Objekt aus Dialog-Konfigurationsoptionen wie von der jQuery UI Library definiert.

### 3.3.8 Typ

funktion `display( row, update, render )`

Parameter:

1. Name: `row`, Typ: `DataTables.Api`, Optional: Nein  
DataTables API-Instanz der gefragten Tabelle, welche vorbevolkert ist mit der Zeile, auf der gearbeitet wird - also dem Ergebnis von `row()`.
2. Name: `update`, Typ: `boolean`, Optional: Nein  
Dieser Parameter informiert die Funktion über den Auslöser des Funktionsaufrufs:
  - `true` - Ein automatisches Update, ausgelöst durch eine Veränderung in der Spaltensichtbarkeit, einem neu Zeichnen der Tabelle oder einer anderen Aktion die nicht der Nutzer ist, der spezifisch die Zeile aktiviert.
  - `false` - Endnutzer hat die Zeile ausgelöst. Dies sollte wahrscheinlich als Schalter (falls die Informationen angezeigt und verborgen werden können) oder als Anzeigeaktion (falls die Informationen nur angezeigt werden können) verwendet werden.
3. Name: `render`, Typ: `funktion`, Optional: Nein  
Die anzuzeigenden Daten - dies wird als Funktion gegeben, damit es nur ausgeführt wird, wenn nötig (die Daten müssen nicht gesammelt werden, falls sie eh nur verborgen werden sollen). Der von dieser Funktion zurückgegebene String ist der, der von der Funktion `responsive.details.renderer` gegeben wird. Akzeptiert keine Parameter.

Rückgabewert:

`true` falls die Funktion die verborgenen Daten angezeigt hat, `false` falls nicht. Diese Information wird verwendet, um die Events auszulösen, die Anzeigen, dass Responsive Informationen angezeigt oder verborgen hat. Ist der Rückgabewert `undefined`, so werden keine Events ausgelöst.

### 3.3.9 Beispiele

Verwende die `childRowImmediate` Anzeigeeoption:

```
$('#example').DataTable( {
  responsive: {
    details: {
      display: $.fn.dataTable.Responsive.display.childRowImmediate,
      type: ''
    }
  }
} );
```

Verwende modal ohne Optionen:

```
$('#example').DataTable( {
  responsive: {
    details: {
      display: $.fn.dataTable.Responsive.display.modal()
    }
  }
} );
```

Verwende modal mit Spezifizierung eines Headers:

```
$('#example').DataTable( {
  responsive: {
    details: {
      display: $.fn.dataTable.Responsive.display.modal( {
        header: function ( row ) {
          var data = row.data();
          return 'Details for '+data.clientName;
        }
      } )
    }
  }
} );
```

table.responsive.details.renderer

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: Funktion, die die verborgenen Informationen in einer ul/li Liste anzeigt. ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Die Informationen in der von Responsive angezeigten Detailzeile werden durch diese Funktion erstellt. Standardmäßig erstellt die Funktion eine ul/li Liste, welche die Daten aus verborgenen Zellen anzeigt, es kann aber auch einer der anderen eingebauten Renderer oder eine eigene Funktion verwendet werden.

Die Renderfunktion wird ausgeführt für die Detailanzeigen in der Tabelle, und läuft immer, wenn sich die Sichtbarkeit der Spalten in der Tabelle verändert.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

Seit v2.1.0 hat Responsive eine Reihe an eingebauten Renderfunktionen, die durch das Objekt `$.fn.dataTable.Responsive.renderer` erreicht werden können. Die eingebauten Renderer liegen als Funktionen vor, die ausgeführt werden müssen und den benötigten Renderer zurückgeben. So können Optionen an den Renderer übergeben werden.

Die eingebauten Optionen sind:

- `listHidden` - Zeige die verborgenen Daten in einer ul / li Liste.
  - Keine zusätzlichen Optionen
  - Standardrenderer
- `tableAll` - Zeige die Daten aller Spalten (egal ob sie verborgen sind oder nicht) in einer Tabelle an.
  - Einzige Option: `tableClass` - Der Klassenname, den die erstellte Tabelle erhalten soll.

Zusätzliche Renderer können erstellt und an dieses Objekt angehängt werden, wenn sie modular sein sollen. Falls ja, können sie frei mit der Community geteilt werden!

Beim Erstellen vom HTML für jedes anzuzeigende Element ist es eine gute Idee, anhand der gegebenen `rowIndex` und `columnIndex` Informationen, die `data-dt-row` und `data-dt-column` Attribute hinzuzufügen. Dadurch können die Elemente und ihre Kinder in die DataTables API Selektormethoden (z.B. `cell()`) übergeben werden und als normaler Teil der API benutzt werden.

### 3.3.10 Typ

`funktion renderer( api, rowIdx, columns )`

Parameter:

1. Name: `api`, Typ: `DataTable.Api`, Optional: Nein  
DataTables API-Instanz der gefragten Tabelle.
2. Name: `rowIdx`, Typ: `integer`, Optional: Nein  
Zeilenindex der Zeile die der Renderer rendern soll. Verwende die `row()` und / oder `cells()` Methoden, um Informationen über die Zeile von der API zu kriegen, damit die Information gerendert werden kann.
3. Name: `columns`, Typ: `Array`, Optional: Nein  
**Seit 2.0.0:** Ein Array aus Objekten mit Informationen über alle Spalten in der Tabelle. Die Länge des Arrays ist gleich der Anzahl der Spalten in der Tabelle und jede Spalte wird in Indexreihenfolge durch ein Objekt repräsentiert. Weiterhin ist die Struktur jedes Objekts:

```

{
  title: "...",           // string - Spaltentitel
  data: "...",           // string - Zellenwert
  hidden: true / false  // boolean - true wenn von Responsive verborgen,
  ansonsten false
  columnIndex: ...     // integer - Spaltendatenindex (seit Responsive
2.0.1)
  rowIndex: ...        // integer - Zeilendatenindex (seit Responsive
2.0.2)
}

```

Rückgabewert:

Zwei Werte können zurückgegeben werden:

- boolean false - Zeige keine Kindzeile an
- string - Die anzuzeigenden Informationen, inklusive allem HTML.

### 3.3.11 Beispiele

Verwende den tableAll-Renderer ohne einen Klassennamen zu spezifizieren:

```

$('#example').DataTable( {
  responsive: {
    details: {
      renderer: $.fn.dataTable.Responsive.renderer.tableAll()
    }
  }
} );

```

Verwende den tableAll-Renderer mit einem spezifizierten Klassennamen:

```

$('#example').DataTable( {
  responsive: {
    details: {
      renderer: $.fn.dataTable.Responsive.renderer.tableAll( {
        tableClass: 'ui table'
      } )
    }
  }
} );

```

Verwende einen eigenen Renderer, welcher die verborgenen Daten in einer HTML-Tabelle anzeigt:

```

$('#example').DataTable( {
  responsive: {
    details: {
      renderer: function ( api, rowIdx, columns ) {
        var data = $.map( columns, function ( col, i ) {
          return col.hidden ?
            '<tr data-dt-row="'+col.rowIndex+'" data-dt-
column="'+col.columnIndex+'>'+
              '<td>'+col.title+':'+</td> '+
              '<td>'+col.data+</td>'+
            '</tr>' :
          '';
        } ).join('');
        return data ?
          $('<table/>').append( data ) :
          false;
      }
    }
  }
} );

```

### table.responsive.details.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inline' ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Responsive hat zwei eingebaute Methoden um zu verwalten, wie die Anzeigekontrolle der Kindzeile dem Endnutzer angezeigt wird:

- inline - Wodurch die erste Spalte der Tabelle zusätzliches Padding kriegt und das Pseudoelement :before verwendet wird, um den Knopf anzuzeigen.
- column - Wodurch eine gesamte Spalte für die Anzeigekontrolle verwendet wird.
- none - Wodurch die Icons für das Anzeigen / Verbergen nicht angezeigt werden. Normalerweise verwendet, wenn der Nutzer die Fähigkeit verliert, Detailzeilen anzuzeigen oder zu verbergen (responsive.details.target).

Der inline-Stil ist nützlich, wenn man bereits eine Tabelle hat, zu der man Responsive hinzufügen will, ohne die Struktur der Tabelle zu verändern, während column nützlich ist, wenn das Layout der ersten Spalte nicht durch das Padding der inline-Variante verändert werden soll.



Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.3.12 Typ

string

Der zu verwendende Anzeigetyp für die Detailzeilen. Mögliche Werte sind:

- inline - Zeige das Kontrollelement in der ersten Spalte an.
- column - Zeige das Kontrollelement in einer eigenen Spalte an. Man beachte, dass, wenn column gewählt wurde und die target-Option auf einem Spaltenindex steht, die Klasse dieser Spalte control sein sollte, damit CSS dem Nutzer eine visuelle Kontrollfläche anzeigen kann.
- none - Zeige keine Icons zum Anzeigen / Verbergen an.

### 3.3.13 Beispiel

Verwende den column-Kontrolltyp:

```
$('#example').DataTable( {
  responsive: {
    details: {
      type: 'column'
    }
  },
  columnDefs: [ {
    className: 'control',
    orderable: false,
    targets: 0
  } ],
  order: [ 1, 'asc' ]
} );
```

Erlaubte Werte

- inline
- column
- none

table.responsive.details.target

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Wenn der Typ `column` in der Option `responsive.details.type` ausgewählt, gibt diese Option die Möglichkeit zu kontrollieren, welches Element in der Tabelle die Zeigen / Verbergen Kontrolle in jeder Zeile aktiviert. Dies kann ein Spaltenindex oder ein jQuery Selektor sein, wie unten beschrieben.

Wenn die `inline` Option für `responsive.details.type` verwendet wird, hat diese Option keinen Effekt.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

### 3.3.14 Typen

#### integer

Spaltenindex, an den die Zeigen / Verbergen Kontrolle angehängt werden soll. Der Wert kann  $\geq 0$  sein, um Spalten von links zu zählen, oder  $< 0$ , um Spalten von rechts zu zählen

Man beachte, dass die hier definierte Spalte die Klasse `control` hinzugefügt bekommen sollte. So kann das CSS von Responsive dem Endnutzer ein Kontrollicon anzeigen.

#### string

Als String wird diese Option als jQuery Selektor verwendet, um zu bestimmen, welche Elemente die Zeigen / Verbergen Kontrolle für die Kindzeilen aktivieren. So kann jedes beliebige Element in der Tabelle verwendet werden - zum Beispiel eine gesamte Zeile oder ein einzelnes `img`-Element in der Zeile.

Wird ein leerer String gegeben, so kann der Nutzer keine Zeilendetails mehr durch Klicken aktivieren.

### 3.3.15 Beispiele

Verwende den `column`-Kontrolltyp mit der Spalte ganz rechts:

```
$('#example').DataTable( {
  responsive: {
    details: {
      type: 'column',
      target: -1
    }
  },
  columnDefs: [ {
    className: 'control',
    orderable: false,
    targets: -1
  } ]
} );
```

Verwende die gesamte Zeile um die Detailzeile anzuzeigen oder zu verbergen:

```
$('#example').DataTable( {
  responsive: {
    details: {
      type: 'column',
      target: 'tr'
    }
  }
} );
```

### table.responsive.orthogonal

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'display' ]

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Wenn Responsive die verborgenen Informationen über eine Tabellenzeile anzeigen soll, verwendet es die Methode `cell().render()`, um auf die Daten jeder Zelle in der Zeile zuzugreifen. Bei dieser Methode können die orthogonalen Daten von DataTables verwendet werden - also die Möglichkeit, gleiche Daten für verschiedene Zwecke verschieden formatiert zu erhalten.

Man betrachte zum Beispiel die Zahl 123456 - für die Anzeige in der Tabelle soll die Zahl vielleicht als Währungswert angezeigt werden: '\$1,234.56', aber für die Suche und Sortierung würde die unformatierte Zahl verwendet werden. Diese Fähigkeit wird dadurch bereitgestellt, dass die Option `columns.render` als Funktion oder Objekt verwendet wird und die zu beschaffenden Daten für jeden Verwendungszweck spezifiziert werden.

Normalerweise soll Responsive einfach die Daten benutzen, wie sie in den Zellen der Tabelle angezeigt werden. Das ist auch die Standardeinstellung, aber vielleicht soll ein bestimmtes anderes Format der Daten für die Anzeige von Responsive verwendet werden. Das wird durch diesen Parameter ermöglicht.

### 3.3.16 Beispiel

Setze einen spezifischen verlangten Datentyp für die Informationsanzeige:

```
$('#example').DataTable( {
  responsive: {
    orthogonal: 'responsive'
  }
} );
```

## Erlaubte Werte

- display
- sort
- filter
- type

## table.responsive

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Responsive
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Man beachte: Dieses Attribut setzt die Responsive-Erweiterung für DataTables voraus.

Diese Option ermöglicht es, Responsive für DataTables zu aktivieren und zu konfigurieren. Einfach als Boolean-Wert `true` gegeben aktiviert diese Option Responsive mit den Standardeinstellungen (die in `$.fn.dataTable.Responsive.defaults` definiert sind). Die Option kann auch als Objekt gegeben werden, um Responsive mit eigenen Konfigurationen zu starten.

Man beachte, dass diese Option, wie alle Konfigurationsoptionen von Responsive, eine Erweiterung der standardmäßigen Optionen von DataTables darstellt. Dieser Parameter sollte im Initialisierungsobjekt der entsprechenden Tabelle gesetzt werden.

## 3.3.17 Typen

### boolean

Als boolean gegeben kann dieser Parameter Responsive für die Tabelle aktivieren, die gerade erstellt wird. `true` aktiviert Responsive, `false` aktiviert Responsive nicht (selbst wenn die Tabelle eine Klasse von `responsive` oder `dt-responsive` hat).

Dies ist eine Abkürzung um Responsive mit der Standardkonfiguration zu aktivieren. Veränderungen an der Konfiguration können vorgenommen werden, indem der Parameter als Objekt übergeben wird. Siehe unten.

## objekt

Als Objekt gegeben wird Responsive für die Tabelle aktiviert, wobei die Standardwerte (\$.fn.dataTable.Responsive.defaults) erweitert und potentiell von den in diesem Objekt gegebenen Optionen überschrieben werden. So kann Responsive für jede Tabelle einzeln konfiguriert werden.

### 3.3.18 Beispiele

Aktiviere Responsive für eine Tabelle:

```
$('#example').DataTable( {  
  responsive: true  
} );
```

Aktiviere Responsive mit eigenen Konfigurationen:

```
$('#example').DataTable( {  
  responsive: {  
    details: false  
  }  
} );
```

## 4 ECharts (2) [Options]

<b>Name</b>	ECharts
<b>Version</b>	2
<b>Hersteller</b>	Baidu
<b>Webseite</b>	<a href="http://echarts.baidu.com/echarts2/index-en.html">http://echarts.baidu.com/echarts2/index-en.html</a>

### 4.1 ECharts (2) Toolbox

#### 4.1.1 chart.toolbox.feature.toggleSeriesVisibility.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	hide-show, serien ausblenden, tradui-custom, visibility

Anmerkung: TRADUI Custom Erweiterung der ECharts Toolbox

Erweitert die ECharts Toolbox um einen Button um alle Serien ein- und auszublenden.  
Nützlich, wenn Sie nur eine bestimmte Serie anzeigen möchten und nicht mehrmals klicken möchten.

## 5 ECharts (4) [Options]

<b>Name</b>	ECharts
<b>Version</b>	4
<b>Hersteller</b>	Apache
<b>Webseite</b>	<a href="https://ecomfe.github.io/echarts-doc/public/en/">https://ecomfe.github.io/echarts-doc/public/en/</a>

### 5.1 ECharts (4) Angle Axis

#### 5.1.1 chart.angleAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Winkelachse im Polarkoordinatensystem.

#### 5.1.2 chart.angleAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.1.3 chart.angleAxis.polarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index der Winkelachse im Polarkoordinatensystem. Standardmäßig wird die erste Achse verwendet.

### 5.1.4 chart.angleAxis.startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 90 ]

Startwinkel der Achse. Standardmäßig 90 Grad, was für von der Mitte aus nach oben zeigend steht. 0 Grad steht für von der Mitte aus nach rechts zeigend.

### 5.1.5 chart.angleAxis.clockwise

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: true ]

Spezifiziert, ob die Werte der Winkelachse im Uhrzeigersinn oder gegen den Uhrzeigersinn ansteigen. Standardmäßig true für im Uhrzeigersinn.

### 5.1.6 chart.angleAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'category' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 5.1.7 chart.angleAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

## 5.1.8 chart.angleAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 5.1.9 chart.angleAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.1.10 chart.angleAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 5.1.11 chart.angleAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.1.12 chart.angleAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.1.13 chart.angleAxis.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den

meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.1.14 chart.angleAxis.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.1.15 chart.angleAxis.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 5.1.16 chart.angleAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

### 5.1.17 chart.angleAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 5.1.18 chart.angleAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

## 5.1.19 chart.angleAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

### Erlaubte Werte

- none
- arrow

## 5.1.20 chart.angleAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.1.21 chart.angleAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.1.22 chart.angleAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.1.23 chart.angleAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]



## Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.1.24 chart.angleAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.1.25 chart.angleAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.1.26 chart.angleAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.27 chart.angleAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.28 chart.angleAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.29 chart.angleAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.30 chart.angleAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.31 chart.angleAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.1.32 chart.angleAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.1.33 chart.angleAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.1.34 chart.angleAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.1.35 chart.angleAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.1.36 chart.angleAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.1.37 chart.angleAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Linienstil der Achsenticks.

### 5.1.38 chart.angleAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Linien der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.1.39 chart.angleAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.1.40 chart.angleAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.1.41 chart.angleAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.42 chart.angleAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.43 chart.angleAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.44 chart.angleAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.45 chart.angleAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.46 chart.angleAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 5.1.47 chart.angleAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.1.48 chart.angleAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.1.49 chart.angleAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.1.50 chart.angleAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.1.51 chart.angleAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.1.52 chart.angleAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.1.53 chart.angleAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.1.54 chart.angleAxis.axisLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {  
  color: function (value, index) {  
    return value >= 0 ? 'green' : 'red';  
  }  
}
```

### 5.1.55 chart.angleAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.1.56 chart.angleAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.1.57 chart.angleAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.1.58 chart.angleAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.1.59 chart.angleAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.1.60 chart.angleAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:



```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.1.61 chart.angleAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.1.62 chart.angleAxis.axisLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.1.63 chart.angleAxis.axisLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.1.64 chart.angleAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.1.65 chart.angleAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.1.66 chart.angleAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.1.67 chart.angleAxis.axisLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.1.68 chart.angleAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.1.69 chart.angleAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.1.70 chart.angleAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.1.71 chart.angleAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.1.72 chart.angleAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.1.73 chart.angleAxis.axisLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.1.74 chart.angleAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.1.75 chart.angleAxis.axisLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.1.76 chart.angleAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.1.77 chart.angleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.1.78 chart.angleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.1.79 chart.angleAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.1.80 chart.angleAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.1.81 chart.angleAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.1.82 chart.angleAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.1.83 chart.angleAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.1.84 chart.angleAxis.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbararray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 5.1.85 chart.angleAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 5.1.86 chart.angleAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.1.87 chart.angleAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.88 chart.angleAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.89 chart.angleAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.90 chart.angleAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.91 chart.angleAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.92 chart.angleAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.1.93 chart.angleAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.1.94 chart.angleAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.1.95 chart.angleAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 5.1.96 chart.angleAxis.splitArea.areaStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.1.97 chart.angleAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.98 chart.angleAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 5.1.99 chart.angleAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.100 chart.angleAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.101 chart.angleAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.102 chart.angleAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 5.1.103 chart.angleAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.1.104 chart.angleAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.1.105 chart.angleAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.1.106 chart.angleAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.1.107 chart.angleAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.1.108 chart.angleAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.1.109 chart.angleAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.1.110 chart.angleAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 5.1.111 chart.angleAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 5.1.112 chart.angleAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.1.113 chart.angleAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 5.1.114 chart.angleAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.1.115 chart.angleAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.1.116 chart.angleAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.1.117 chart.angleAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.1.118 chart.angleAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.1.119 chart.angleAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.1.120 chart.angleAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.1.121 chart.angleAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.1.122 chart.angleAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.1.123 chart.angleAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.1.124 chart.angleAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.1.125 chart.angleAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.1.126 chart.angleAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.1.127 chart.angleAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.1.128 chart.angleAxis.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.1.129 chart.angleAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.1.130 chart.angleAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.131 chart.angleAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.132 chart.angleAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.133 chart.angleAxis.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.134 chart.angleAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.1.135 chart.angleAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

### 5.1.136 chart.angleAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.1.137 chart.angleAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.1.138 chart.angleAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.139 chart.angleAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.140 chart.angleAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.141 chart.angleAxis.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.142 chart.angleAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.143 chart.angleAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.1.144 chart.angleAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.1.145 chart.angleAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 5.1.146 chart.angleAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 5.1.147 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.148 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.149 chart.angleAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.1.150 chart.angleAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.1.151 chart.angleAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.1.152 chart.angleAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 5.1.153 chart.angleAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.1.154 chart.angleAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 5.1.155 chart.angleAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.1.156 chart.angleAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.1.157 chart.angleAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.1.158 chart.angleAxis.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.1.159 chart.angleAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.1.160 chart.angleAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.1.161 chart.angleAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.1.162 chart.angleAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.1.163 chart.angleAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.1.164 chart.angleAxis.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.1.165 chart.angleAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

## 5.2 ECharts (4) Axis

### 5.2.1 chart.yAxis.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Y-Achse in kartesischen (rechteckigen) Koordinatensystemen. Normalerweise kann eine einzelne Gitterkomponente höchstens 2 Y-Achsen platzieren, eine links und eine rechts. Ein Offset kann verwendet werden, um Überlappung bei mehr als zwei Y-Achsen zu vermeiden.

### 5.2.2 chart.yAxis.{x}.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.2.3 chart.yAxis.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Y-Achse angezeigt werden soll.

### 5.2.4 chart.yAxis.{x}.gridIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index des Gitters zu dem die Y-Achse gehört. Standardmäßig wird das erste Gitter genommen.

### 5.2.5 chart.yAxis.{x}.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position der Y-Achse.

Die erste Y-Achse im Gitter wird standardmäßig auf der linken Seite ('left') platziert, und die zweite auf der gegenüberliegenden Seite zur ersten Y-Achse.



## Erlaubte Werte

- left
- right

### 5.2.6 chart.yAxis.{x}.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset der Y-Achse relativ zur Standardposition. Nützlich wenn mehrere Y-Achsen den gleichen Positionswert haben.

### 5.2.7 chart.yAxis.{x}.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

## Erlaubte Werte

- value
- category
- time
- log

### 5.2.8 chart.yAxis.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

### 5.2.9 chart.yAxis.{x}.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

## Erlaubte Werte

- start
- middle
- center
- end

### 5.2.10 chart.yAxis.{x}.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 5.2.11 chart.yAxis.{x}.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 5.2.12 chart.yAxis.{x}.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

## Erlaubte Werte

- normal
- italic
- oblique

### 5.2.13 chart.yAxis.{x}.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.2.14 chart.yAxis.{x}.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.2.15 chart.yAxis.{x}.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Axis-Name Schriftgröße.

### 5.2.16 chart.yAxis.{x}.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.2.17 chart.yAxis.{x}.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.2.18 chart.yAxis.{x}.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.19 chart.yAxis.{x}.nameTextStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.2.20 chart.yAxis.{x}.nameTextStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.2.21 chart.yAxis.{x}.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.2.22 chart.yAxis.{x}.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.2.23 chart.yAxis.{x}.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.2.24 chart.yAxis.{x}.nameTextStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.2.25 chart.yAxis.{x}.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.2.26 chart.yAxis.{x}.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.2.27 chart.yAxis.{x}.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.2.28 chart.yAxis.{x}.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.29 chart.yAxis.{x}.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.30 chart.yAxis.{x}.nameTextStyle.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.2.31 chart.yAxis.{x}.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.2.32 chart.yAxis.{x}.nameTextStyle.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.33 chart.yAxis.{x}.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.34 chart.yAxis.{x}.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.35 chart.yAxis.{x}.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.2.36 chart.yAxis.{x}.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.2.37 chart.yAxis.{x}.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Lücke zwischen Achsenname und Achsenlinie.

### 5.2.38 chart.yAxis.{x}.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 5.2.39 chart.yAxis.{x}.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist.

### 5.2.40 chart.yAxis.{x}.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 5.2.41 chart.yAxis.{x}.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.2.42 chart.yAxis.{x}.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 5.2.43 chart.yAxis.{x}.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

## 5.2.44 chart.yAxis.{x}.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.2.45 chart.yAxis.{x}.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.2.46 chart.yAxis.{x}.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.2.47 chart.yAxis.{x}.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.2.48 chart.yAxis.{x}.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 5.2.49 chart.yAxis.{x}.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 5.2.50 chart.yAxis.{x}.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

## 5.2.51 chart.yAxis.{x}.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.2.52 chart.yAxis.{x}.axisLine.onZero

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X- oder Y-Achse auf dem Ursprung der jeweils anderen liegt, wo der Achsenwert 0 ist. Nur gültig, wenn die andere Achse den Typ 'value' hat und den Wert 0 enthält.

### 5.2.53 chart.yAxis.{x}.axisLine.onZeroAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere Achsen existieren, kann dieser Wert verwendet werden, um die Achse zu spezifizieren, auf die sich 'onZero' bezieht.

### 5.2.54 chart.yAxis.{x}.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.2.55 chart.yAxis.{x}.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.2.56 chart.yAxis.{x}.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.2.57 chart.yAxis.{x}.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.2.58 chart.yAxis.{x}.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.2.59 chart.yAxis.{x}.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.2.60 chart.yAxis.{x}.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 5.2.61 chart.yAxis.{x}.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.62 chart.yAxis.{x}.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.63 chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.64 chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.65 chart.yAxis.{x}.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.66 chart.yAxis.{x}.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.2.67 chart.yAxis.{x}.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.2.68 chart.yAxis.{x}.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.2.69 chart.yAxis.{x}.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.70 chart.yAxis.{x}.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.2.71 chart.yAxis.{x}.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.2.72 chart.yAxis.{x}.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.2.73 chart.yAxis.{x}.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.2.74 chart.yAxis.{x}.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.2.75 chart.yAxis.{x}.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.76 chart.yAxis.{x}.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.77 chart.yAxis.{x}.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 5.2.78 chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.79 chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.80 chart.yAxis.{x}.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.81 chart.yAxis.{x}.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenbeschriftung gehören.

### 5.2.82 chart.yAxis.{x}.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.2.83 chart.yAxis.{x}.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 5.2.84 chart.yAxis.{x}.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.2.85 chart.yAxis.{x}.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.2.86 chart.yAxis.{x}.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.2.87 chart.yAxis.{x}.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.2.88 chart.yAxis.{x}.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.2.89 chart.yAxis.{x}.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.2.90 chart.yAxis.{x}.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.2.91 chart.yAxis.{x}.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.2.92 chart.yAxis.{x}.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.2.93 chart.yAxis.{x}.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.2.94 chart.yAxis.{x}.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.2.95 chart.yAxis.{x}.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.2.96 chart.yAxis.{x}.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.2.97 chart.yAxis.{x}.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.98 chart.yAxis.{x}.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.2.99 chart.yAxis.{x}.axisLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.2.100 chart.yAxis.{x}.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.2.101 chart.yAxis.{x}.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.2.102 chart.yAxis.{x}.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.2.103 chart.yAxis.{x}.axisLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.2.104 chart.yAxis.{x}.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.2.105 chart.yAxis.{x}.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.2.106 chart.yAxis.{x}.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.2.107 chart.yAxis.{x}.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.108 chart.yAxis.{x}.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.109 chart.yAxis.{x}.axisLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.2.110 chart.yAxis.{x}.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.2.111 chart.yAxis.{x}.axisLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.112 chart.yAxis.{x}.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.113 chart.yAxis.{x}.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.114 chart.yAxis.{x}.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.2.115 chart.yAxis.{x}.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.2.116 chart.yAxis.{x}.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.2.117 chart.yAxis.{x}.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.2.118 chart.yAxis.{x}.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.119 chart.yAxis.{x}.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.2.120 chart.yAxis.{x}.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 5.2.121 chart.yAxis.{x}.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 5.2.122 chart.yAxis.{x}.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.123 chart.yAxis.{x}.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.124 chart.yAxis.{x}.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.125 chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.126 chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.127 chart.yAxis.{x}.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.128 chart.yAxis.{x}.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.2.129 chart.yAxis.{x}.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.130 chart.yAxis.{x}.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.2.131 chart.yAxis.{x}.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 5.2.132 chart.yAxis.{x}.splitArea.areaStyle.color

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.2.133 chart.yAxis.{x}.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.134 chart.yAxis.{x}.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.135 chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.136 chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.137 chart.yAxis.{x}.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.138 chart.yAxis.{x}.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 5.2.139 chart.yAxis.{x}.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.2.140 chart.yAxis.{x}.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.2.141 chart.yAxis.{x}.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.2.142 chart.yAxis.{x}.axisPointer.z

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.2.143 chart.yAxis.{x}.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.2.144 chart.yAxis.{x}.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.2.145 chart.yAxis.{x}.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.2.146 chart.yAxis.{x}.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

## 5.2.147 chart.yAxis.{x}.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 5.2.148 chart.yAxis.{x}.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.2.149 chart.yAxis.{x}.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 5.2.150 chart.yAxis.{x}.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.2.151 chart.yAxis.{x}.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.2.152 chart.yAxis.{x}.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.2.153 chart.yAxis.{x}.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.154 chart.yAxis.{x}.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.155 chart.yAxis.{x}.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.156 chart.yAxis.{x}.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.2.157 chart.yAxis.{x}.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.2.158 chart.yAxis.{x}.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.159 chart.yAxis.{x}.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.160 chart.yAxis.{x}.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.161 chart.yAxis.{x}.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.2.162 chart.yAxis.{x}.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.2.163 chart.yAxis.{x}.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.2.164 chart.yAxis.{x}.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.2.165 chart.yAxis.{x}.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.2.166 chart.yAxis.{x}.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.167 chart.yAxis.{x}.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.168 chart.yAxis.{x}.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.169 chart.yAxis.{x}.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.170 chart.yAxis.{x}.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 5.2.171 chart.yAxis.{x}.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.2.172 chart.yAxis.{x}.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.2.173 chart.yAxis.{x}.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.174 chart.yAxis.{x}.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.175 chart.yAxis.{x}.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.176 chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.177 chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.178 chart.yAxis.{x}.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.179 chart.yAxis.{x}.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.2.180 chart.yAxis.{x}.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.2.181 chart.yAxis.{x}.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.182 chart.yAxis.{x}.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.183 chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.184 chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.185 chart.yAxis.{x}.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.186 chart.yAxis.{x}.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.2.187 chart.yAxis.{x}.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.2.188 chart.yAxis.{x}.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status.

Erlaubte Werte

- show
- hide

### 5.2.189 chart.yAxis.{x}.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der `axisPointer` gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.2.190 `chart.yAxis.{x}.axisPointer.handle.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 5.2.191 `chart.yAxis.{x}.axisPointer.handle.icon`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über `'path://'` auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.2.192 chart.yAxis.{x}.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.2.193 chart.yAxis.{x}.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.2.194 chart.yAxis.{x}.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.2.195 chart.yAxis.{x}.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.2.196 chart.yAxis.{x}.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.197 chart.yAxis.{x}.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.198 chart.yAxis.{x}.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.199 chart.yAxis.{x}.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.200 chart.yAxis.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente der Y-Achse.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.2.201 chart.yAxis.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente der Y-Achse, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.2.202 chart.xAxis.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die X-Achse in kartesischen (rechteckigen) Koordinatensystemen. Normalerweise kann eine einzelne Gitterkomponente höchstens 2 X-Achsen platzieren, eine links und eine rechts. Ein Offset kann verwendet werden, um Überlappung bei mehr als zwei X-Achsen zu vermeiden.

### 5.2.203 chart.xAxis.{x}.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.2.204 chart.xAxis.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X-Achse angezeigt werden soll.

### 5.2.205 chart.xAxis.{x}.gridIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index des Gitters zu dem die X-Achse gehört. Standardmäßig wird das erste Gitter genommen.

### 5.2.206 chart.xAxis.{x}.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position der X-Achse.

Die erste X-Achse im Gitter wird standardmäßig unten platziert, und die zweite auf der gegenüberliegenden Seite zur ersten X-Achse.

#### Erlaubte Werte

- top
- bottom



### 5.2.207 chart.xAxis.{x}.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset der X-Achse relativ zur Standardposition. Nützlich wenn mehrere X-Achsen den gleichen Positionswert haben.

### 5.2.208 chart.xAxis.{x}.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheiden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 5.2.209 chart.xAxis.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

### 5.2.210 chart.xAxis.{x}.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 5.2.211 chart.xAxis.{x}.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 5.2.212 chart.xAxis.{x}.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 5.2.213 chart.xAxis.{x}.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.2.214 chart.xAxis.{x}.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.2.215 chart.xAxis.{x}.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie des Achsennamens.

Akzeptiert auch 'serif', 'monospace', ...

### 5.2.216 chart.xAxis.{x}.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 5.2.217 chart.xAxis.{x}.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.2.218 chart.xAxis.{x}.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.2.219 chart.xAxis.{x}.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.220 chart.xAxis.{x}.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.2.221 chart.xAxis.{x}.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.2.222 chart.xAxis.{x}.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.2.223 chart.xAxis.{x}.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.



### 5.2.224 chart.xAxis.{x}.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.2.225 chart.xAxis.{x}.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.2.226 chart.xAxis.{x}.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.2.227 chart.xAxis.{x}.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.2.228 chart.xAxis.{x}.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.2.229 chart.xAxis.{x}.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.230 chart.xAxis.{x}.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.231 chart.xAxis.{x}.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.2.232 chart.xAxis.{x}.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.2.233 chart.xAxis.{x}.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.234 chart.xAxis.{x}.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.235 chart.xAxis.{x}.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.236 chart.xAxis.{x}.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.2.237 chart.xAxis.{x}.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.2.238 chart.xAxis.{x}.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 5.2.239 chart.xAxis.{x}.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 5.2.240 chart.xAxis.{x}.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 5.2.241 chart.xAxis.{x}.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 5.2.242 chart.xAxis.{x}.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.



Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.2.243 chart.xAxis.{x}.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.2.244 chart.xAxis.{x}.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 5.2.245 chart.xAxis.{x}.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.2.246 chart.xAxis.{x}.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.2.247 chart.xAxis.{x}.interval

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.2.248 `chart.xAxis.{x}.logBase`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.2.249 `chart.xAxis.{x}.silent`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 5.2.250 chart.xAxis.{x}.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 5.2.251 chart.xAxis.{x}.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Einstellungen, die zu axisLine gehören.

### 5.2.252 chart.xAxis.{x}.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.2.253 chart.xAxis.{x}.axisLine.onZero

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X- oder Y-Achse auf dem Ursprung der jeweils anderen liegt, wo der Achsenwert 0 ist. Nur gültig, wenn die andere Achse den Typ 'value' hat und den Wert 0 enthält.

### 5.2.254 chart.xAxis.{x}.axisLine.onZeroAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere Achsen existieren, kann dieser Wert verwendet werden, um die Achse zu spezifizieren, auf die sich 'onZero' bezieht.

### 5.2.255 chart.xAxis.{x}.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.2.256 chart.xAxis.{x}.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.2.257 chart.xAxis.{x}.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.2.258 chart.xAxis.{x}.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.2.259 chart.xAxis.{x}.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```



### 5.2.260 chart.xAxis.{x}.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.2.261 chart.xAxis.{x}.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

### 5.2.262 chart.xAxis.{x}.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.263 chart.xAxis.{x}.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.264 chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.265 chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.266 chart.xAxis.{x}.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.267 chart.xAxis.{x}.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Einstellungen, die zu den Achsenticks gehören.

### 5.2.268 chart.xAxis.{x}.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.2.269 chart.xAxis.{x}.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.2.270 chart.xAxis.{x}.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.271 chart.xAxis.{x}.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.2.272 chart.xAxis.{x}.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.2.273 chart.xAxis.{x}.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.2.274 chart.xAxis.{x}.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.2.275 chart.xAxis.{x}.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.2.276 chart.xAxis.{x}.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.277 chart.xAxis.{x}.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.278 chart.xAxis.{x}.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.279 chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.280 chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.281 chart.xAxis.{x}.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
-----------------	---------



<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.282 chart.xAxis.{x}.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 5.2.283 chart.xAxis.{x}.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.2.284 chart.xAxis.{x}.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.285 chart.xAxis.{x}.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.2.286 chart.xAxis.{x}.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.2.287 chart.xAxis.{x}.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.2.288 chart.xAxis.{x}.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```

// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}

```

### 5.2.289 chart.xAxis.{x}.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.2.290 chart.xAxis.{x}.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.2.291 chart.xAxis.{x}.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.2.292 chart.xAxis.{x}.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.2.293 chart.xAxis.{x}.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.2.294 chart.xAxis.{x}.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.2.295 chart.xAxis.{x}.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.2.296 chart.xAxis.{x}.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.2.297 chart.xAxis.{x}.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom



### 5.2.298 chart.xAxis.{x}.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.299 chart.xAxis.{x}.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.2.300 chart.xAxis.{x}.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.2.301 chart.xAxis.{x}.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.2.302 chart.xAxis.{x}.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.2.303 chart.xAxis.{x}.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.2.304 chart.xAxis.{x}.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.2.305 chart.xAxis.{x}.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.2.306 chart.xAxis.{x}.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.2.307 chart.xAxis.{x}.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.2.308 chart.xAxis.{x}.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.309 chart.xAxis.{x}.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.2.310 chart.xAxis.{x}.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 5.2.311 chart.xAxis.{x}.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 5.2.312 chart.xAxis.{x}.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.313 chart.xAxis.{x}.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.314 chart.xAxis.{x}.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.315 chart.xAxis.{x}.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.2.316 chart.xAxis.{x}.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.2.317 chart.xAxis.{x}.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.2.318 chart.xAxis.{x}.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.2.319 chart.xAxis.{x}.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.320 chart.xAxis.{x}.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.2.321 chart.xAxis.{x}.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 5.2.322 chart.xAxis.{x}.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 5.2.323 chart.xAxis.{x}.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.324 chart.xAxis.{x}.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.325 chart.xAxis.{x}.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.326 chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.327 chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.328 chart.xAxis.{x}.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.329 chart.xAxis.{x}.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.2.330 chart.xAxis.{x}.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.2.331 chart.xAxis.{x}.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.2.332 chart.xAxis.{x}.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 5.2.333 chart.xAxis.{x}.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbararray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.2.334 chart.xAxis.{x}.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.335 chart.xAxis.{x}.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 5.2.336 chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.337 chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.338 chart.xAxis.{x}.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.339 chart.xAxis.{x}.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 5.2.340 chart.xAxis.{x}.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.2.341 chart.xAxis.{x}.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.2.342 chart.xAxis.{x}.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.2.343 chart.xAxis.{x}.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.2.344 chart.xAxis.{x}.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.2.345 chart.xAxis.{x}.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.2.346 chart.xAxis.{x}.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.2.347 chart.xAxis.{x}.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.2.348 chart.xAxis.{x}.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 5.2.349 chart.xAxis.{x}.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.2.350 chart.xAxis.{x}.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 5.2.351 chart.xAxis.{x}.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.2.352 chart.xAxis.{x}.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.2.353 chart.xAxis.{x}.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.2.354 chart.xAxis.{x}.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.2.355 chart.xAxis.{x}.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.356 chart.xAxis.{x}.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.2.357 chart.xAxis.{x}.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.2.358 chart.xAxis.{x}.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.2.359 chart.xAxis.{x}.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.2.360 chart.xAxis.{x}.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.2.361 chart.xAxis.{x}.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.2.362 chart.xAxis.{x}.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.2.363 chart.xAxis.{x}.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.2.364 chart.xAxis.{x}.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.2.365 chart.xAxis.{x}.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.2.366 chart.xAxis.{x}.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.2.367 chart.xAxis.{x}.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.368 chart.xAxis.{x}.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.369 chart.xAxis.{x}.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.370 chart.xAxis.{x}.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.371 chart.xAxis.{x}.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 5.2.372 chart.xAxis.{x}.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.2.373 chart.xAxis.{x}.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.2.374 chart.xAxis.{x}.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.2.375 chart.xAxis.{x}.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.376 chart.xAxis.{x}.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.377 chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.378 chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.379 chart.xAxis.{x}.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.380 chart.xAxis.{x}.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.2.381 chart.xAxis.{x}.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.2.382 chart.xAxis.{x}.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.383 chart.xAxis.{x}.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.384 chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.385 chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.386 chart.xAxis.{x}.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.2.387 chart.xAxis.{x}.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.2.388 chart.xAxis.{x}.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.2.389 chart.xAxis.{x}.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status.

Erlaubte Werte

- show
- hide

### 5.2.390 chart.xAxis.{x}.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.2.391 chart.xAxis.{x}.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 5.2.392 chart.xAxis.{x}.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.2.393 chart.xAxis.{x}.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.2.394 chart.xAxis.{x}.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.2.395 chart.xAxis.{x}.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.2.396 chart.xAxis.{x}.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.2.397 chart.xAxis.{x}.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.2.398 chart.xAxis.{x}.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.2.399 chart.xAxis.{x}.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.2.400 chart.xAxis.{x}.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.2.401 chart.xAxis.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente der X-Achse.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.2.402 chart.xAxis.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente der X-Achse, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

## 5.3 ECharts (4) Axis Pointer

### 5.3.1 chart.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dies ist die globale Konfigurierung von axisPointer.

axisPointer ist ein Werkzeug für die Darstellung von Referenzlinien und Achsenwerten unter dem Mauszeiger.

axisPointer kann auch auf Geräten mit Berührungssteuerung verwendet werden, wo Nutzer einen Knopf ziehen können, um Referenzlinien und Achsenwerte zu bewegen.

In Fällen in denen mehr als eine Achse existiert, hilft axisPointer, in die Daten zu sehen.

**Hinweis:** Im Allgemeinen wird der axisPointer für jede Achse die ihn braucht konfiguriert (zum Beispiel in xAxis.axisPointer) oder im Tooltip konfiguriert (zum Beispiel in tooltip.axisPointer).

Die folgenden Konfigurationen gibt es jedoch nur im globalen axisPointer: axisPointer.triggerOn, axisPointer.link.

#### Wie man axisPointer anzeigt:

In kartesischen (grid), polaren (polar) und Einzelachsen-Diagrammen (single axis) hat jede Achse ihren eigenen axisPointer.

Diese axisPointer werden standardmäßig nicht angezeigt, bis sie wie folgt konfiguriert wurden:

- Man setzt eineAchse.axisPointer.show (z.B. xAxis.axisPointer.show) auf true. Dann wird der axisPointer dieser Achse angezeigt.
- Man setzt tooltip.trigger auf 'axis' oder tooltip.axisPointer.type auf 'cross'. Dann wählt das Koordinatensystem automatisch die Achsen, die ihren axisPointer anzeigen. (tooltip.axisPointer.axis kann verwendet werden, um die Entscheidung zu ändern.) Es ist anzumerken, dass axis.axisPointer die Einstellungen von tooltip.axisPointer überschreibt.

#### Wie man die Beschriftung von axisPointer anzeigt:

Die Beschriftung des axisPointers wird standardmäßig nicht angezeigt (nur die Referenzlinie wird standardmäßig angezeigt), bis sie wie folgt konfiguriert wurde:

- Man setzt eineAchse.axisPointer.label.show (z.B. xAxis.axisPointer.label.show) auf true. Dann wird die Beschriftung des axisPointers angezeigt.

- Man setzt `tooltip.axisPointer.type` auf `'cross'`. Dann werden die Beschriftungen der gekreuzten `axisPointer` angezeigt.

#### Wie man `axisPointer` auf Geräten mit Berührungssteuerung konfiguriert:

Man setzt eine Achse `axisPointer.handle.show` (z.B. `xAxis.axisPointer.handle.show`) auf `true`. Dann wird der ziehbare Knopf angezeigt. (Diese Feature wird für polar nicht unterstützt.)

**Hinweis:** Funktioniert der Tooltip in diesem Fall nicht gut, kann man versuchen, `tooltip.triggerOn` auf `'none'` zu setzen (für den Effekt: Zeige den Tooltip wenn der Finger den Knopf hält und verstecke ihn wenn der Finger den Knopf loslässt) oder `tooltip.alwaysShowContent` auf `true` zu setzen (dann wird der Tooltip immer angezeigt).

#### Auf Punkte einrasten:

Bei Zeit- und kontinuierlichen Achsen rastet der `axisPointer` automatisch auf Datenpunkten ein, falls `snap` auf `true` gesetzt wurde.

### 5.3.2 `chart.axisPointer.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.3.3 `chart.axisPointer.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: `false` ]

Der `axisPointer` wird standardmäßig nicht angezeigt. Wurde `tooltip.trigger` jedoch auf `'axis'` oder `tooltip.axisPointer.type` auf `'cross'` gesetzt, so wird der `axisPointer` automatisch angezeigt. Jedes



Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.3.4 chart.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatorotyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.3.5 chart.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.3.6 chart.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.3.7 chart.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.3.8 chart.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.3.9 chart.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.3.10 chart.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.3.11 chart.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 5.3.12 chart.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.3.13 chart.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'

- 'oblique'

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.3.14 chart.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default -Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

#### Erlaubte Werte

- norma
- bold
- bolder
- lighter
- {integer}

### 5.3.15 chart.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.3.16 chart.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.3.17 chart.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.3.18 chart.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.3.19 chart.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.3.20 chart.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.3.21 chart.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.3.22 chart.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.3.23 chart.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.3.24 chart.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.3.25 chart.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.3.26 chart.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.3.27 chart.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig gleich zu axis.axisLine.lineStyle.color .

### 5.3.28 chart.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.3.29 chart.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite der Beschriftung.

### 5.3.30 chart.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.3.31 chart.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.3.32 chart.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.3.33 chart.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.3.34 chart.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type auf 'line' steht.

### 5.3.35 chart.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.3.36 chart.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 5.3.37 chart.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Mögliche Wert:

- 'solid'
- 'dashed'
- 'dotted'

Erlaubte Werte

- solid
- dotted
- dashed

### 5.3.38 chart.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.3.39 chart.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.3.40 chart.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.3.41 chart.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.3.42 chart.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.3.43 chart.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.3.44 chart.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.3.45 chart.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.3.46 chart.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.3.47 chart.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.3.48 chart.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.3.49 chart.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.3.50 chart.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.3.51 chart.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.3.52 chart.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 5.3.53 chart.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.3.54 chart.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um den Handle zu verwenden.

### 5.3.55 chart.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon des Handles.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.3.56 chart.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.3.57 chart.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.3.58 chart.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.3.59 chart.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.3.60 chart.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.3.61 chart.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.3.62 chart.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.3.63 chart.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.3.64 chart.axisPointer.link

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

axisPointer können miteinander verlinkt werden. Der Begriff "link" zeigt an, dass Achsen miteinander synchronisiert sind und sich zusammen bewegen. Achsen Achsen werden anhand des Wertes von axisPointer verlinkt.

link ist ein Array, in dem jedes Element eine "link group" repräsentiert. Achsen werden verlinkt wenn sie in der selben link group referenziert werden. Zum Beispiel:

```
link: [
  {
    // Alle Achsen mit xAxisIndex 0, 3, 4 und yAxisName "sameName" werden
    verlinkt
    xAxisIndex: [0, 3, 4],
    yAxisName: 'sameName'
  },
  {
    // Alle Achsen mit xAxisId 'aa', 'cc' und alle angleAxis werden verlinkt.
    xAxisId: ['aa', 'cc'],
    angleAxis: 'all'
  },
  ...
]
```

Wie oben gezeigt können Achsen in einer link group auf diese Arten referenziert werden:

```
{
  // 'some' steht für den Dimensionsnamen einer Achse, namentlich 'x', 'y',
  // 'radius', 'angle' oder 'single'
  someAxisIndex: [...], // kann ein Array, Wert oder 'all' sein
  someAxisName: [...], // kann ein Array, Wert oder 'all' sein
  someAxisId: [...],    // kann ein Array, Wert oder 'all' sein
}
```

### Wie verlinkt man Achsen mit verschiedenem axis.type?

Zum Beispiel, der Typ von axisA ist 'category' und der Typ von axisB ist 'time'. Wir können eine Konvertierungsfunktion (Mapper) in der link group schreiben um Werte von einer Achse zur anderen zu konvertieren. zum Beispiel:

```
link: [{
  xAxisIndex: [0, 1],
  yAxisName: ['yy'],
  mapper: function (sourceVal, sourceAxisInfo, targetAxisInfo) {
    if (sourceAxisInfo.axisName === 'yy') {
      // von Zeitstempel zu '2012-02-05'
      return echarts.format.formatTime('yyyy-MM-dd', sourceVal);
    }
    else if (targetAxisInfo.axisName === 'yy') {
      // von '2012-02-05' zu date
      return echarts.number.parseDate(dates[sourceVal]);
    }
    else {
      return sourceVal;
    }
  }
}]
```

Eingabeparameter von Mapper:

{number} sourceVal

{Object} sourceAxisInfo enthält {axisDim, axisId, axisName, axisIndex, ...}

{Object} targetAxisInfo enthält {axisDim, axisId, axisName, axisIndex, ...}

Rückgabewert von Mapper:

{number} Das Ergebnis der Konvertierung

### 5.3.65 chart.axisPointer.triggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mousemove|click' ]

Konditionen unter denen der Tooltip auslöst. Mögliche Werte:

- 'mousemove'  
Löse aus wenn die Maus sich bewegt.
- 'click'  
Löse aus wenn die Maus klickt.
- 'mousemove|click'  
Löse aus wenn die Maus klickt und sich bewegt.
- 'none'  
Löse nicht durch bewegen und klicken der Maus aus.

Erlaubte Werte

- mousemove
- click
- mousemove|click
- none

## 5.4 ECharts (4) Brush

### 5.4.1 chart.brush.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

brush ist eine Komponente zum Auswählen von Bereichen, mit der Nutzer Teile der Daten eines Diagramms auswählen können, um diese im Detail zu betrachten oder Berechnungen mit ihnen durchzuführen.

### 5.4.2 chart.brush.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.4.3 chart.brush.toolbox

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rect', 'polygon', 'keep', 'clear'] ]

In der Toolbox zu verwendende Knöpfe.

Knöpfe der Toolbox die zu brush gehören sind:

- 'rect': Für rechteckige Auswahlbox;
- 'polygon': Für vieleckige Auswahlbox;
- 'lineX': Für horizontale Auswahlbox;
- 'lineY': Für vertikale Auswahlbox;
- 'keep': Für das Wechseln zwischen einzelner und multipler Auswahl. Die Erstere unterstützt das Löschen des Auswahlbereichs beim Mausklick, die Letztere unterstützt mehrere Auswahlbereiche.
- 'clear': Für das Löschen aller Auswahlbereiche

#### Erlaubte Werte

- rect
- polygon
- lineX
- lineY
- keep
- clear

### 5.4.4 chart.brush.brushLink

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Verbindet Interaktionen zwischen ausgewählten Elementen in verschiedenen Serien.

Es folgt ein Beispiel für das Aktivieren vom ausgewählten Effekt für Streu- und Paralleldiagramme sobald ein Streudiagramm ausgewählt ist.

brushLink ist ein Array aus seriesIndex-Werten, welches die Serien bestimmt, deren Interaktionen verbunden sind. Der Wert kann zum Beispiel sein:

- [3, 4, 5] für Interaktionen zwischen den Serien mit seriesIndex 3, 4 oder 5;
- 'all' für Interaktionen zwischen allen Serien;
- 'none', null oder undefined um brushLink zu deaktivieren.

#### Achtung

brushLink ist ein Mapping von dataIndex. **Also sollten die Daten aller mit brushLink verbundenen Serien garantiert zueinander korrespondieren.**

Beispiel:

```
option = {
  brush: {
    brushLink: [0, 1]
  },
  series: [
    {
      type: 'bar'
      data: [232, 4434, 545, 654] // data hat 4 Einträge
    },
    {
      type: 'parallel',
      data: [[4, 5], [3, 5], [66, 33], [99, 66]] // data hat auch 4 Einträge,
      was zu dem data oben korrespondiert
    }
  ]
};
```

#### 5.4.5 chart.brush.seriesIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'all' ]

Legt fest, welche der Serien brush-Auswahl verwenden kann. Der Wert kann sein:

- 'all': Alle Serien;
- Array: Array an Serienindices
- Nummer: Bestimmter Serienindex

### 5.4.6 chart.brush.geoIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der geo-Diagramme die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

#### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

#### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von brush.geoIndex, brush.xAxisIndex und brush.yAxisIndex festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:

```
option = {
  geo: {
    ...
  },
  brush: {
    geoIndex: 'all', // brush-Auswahl ist nur in den oberen geo-Diagrammen
    aktiviert
    ...
  }
};
```

### 5.4.7 chart.brush.xAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der xAxisIndex die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

#### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

#### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von brush.geoIndex, brush.xAxisIndex und brush.yAxisIndex festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:

```

option = {
  grid: [
    {...}, // grid 0
    {...} // grid 1
  ],
  xAxis: [
    {gridIndex: 1, ...}, // xAxis 0 für grid 1
    {gridIndex: 0, ...} // xAxis 1 für grid 0
  ],
  yAxis: [
    {gridIndex: 1, ...}, // yAxis 0 für grid 1
    {gridIndex: 0, ...} // yAxis 1 für grid 0
  ],
  brush: {
    xAxisIndex: [0, 1], // Brush-Auswahl ist nur für Koordinaten mit xAxisIndex 0
oder 1 aktiviert
    ...
  }
};

```

### 5.4.8 chart.brush.yAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der yAxisIndex die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

#### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

#### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von `brush.geoIndex`, `brush.xAxisIndex` und `brush.yAxisIndex` festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:

```
option = {
  grid: [
    {...}, // grid 0
    {...} // grid 1
  ],
  xAxis: [
    {gridIndex: 1, ...}, // xAxis 0 für grid 1
    {gridIndex: 0, ...} // xAxis 1 für grid 0
  ],
  yAxis: [
    {gridIndex: 1, ...}, // yAxis 0 für grid 1
    {gridIndex: 0, ...} // yAxis 1 für grid 0
  ],
  brush: {
    yAxisIndex: [0, 1], // Brush-Auswahl ist nur für Koordinaten mit yAxisIndex 0
    oder 1 aktiviert
    ...
  }
};
```

### 5.4.9 chart.brush.brushType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rect' ]

Standardtyp der Auswahl.

- 'rect': Für eine rechteckige Auswahlbox;
- 'polygon': Für eine vieleckige Auswahlbox;
- 'lineX': Für eine horizontale Auswahlbox;
- 'lineY': Für eine vertikale Auswahlbox;

## Erlaubte Werte

- rect
- polygon
- lineX
- lineY

### 5.4.10 chart.brush.brushMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'single' ]

Standardmäßiger brush-Modus, dessen Wert sein kann:

- 'single': Für Einzelauswahl;
- 'multiple': Für Mehrfachauswahl.

## Erlaubte Werte

- single
- multiple

### 5.4.11 chart.brush.transformable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Bestimmt, ob eine ausgewählte Box in ihrer Form verändert oder übersetzt werden kann.

### 5.4.12 chart.brush.brushStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Standardmäßiger brush-Stil, dessen Wert ist:

```
{
  borderWidth: 1,
  color: 'rgba(120,140,180,0.3)',
  borderColor: 'rgba(120,140,180,0.8)'
},
```

### 5.4.13 chart.brush.throttleType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'fixRate' ]

Standardmäßig wird brushSelected ausgelöst wann immer eine Auswahlbox ausgewählt oder bewegt wird, um die Umgebung über das Event zu informieren.

Zu häufiges Auslösen des Events kann jedoch zu Effizienzproblemen oder Beeinträchtigung der Animation führen. Deshalb stellt die brush-Komponente brush.throttleType und brush.throttleDelay zur Verfügung, um dieses Problem zu lösen.

Gültige Werte für throttleType sind:

- 'debounce': Um das Event erst auszulösen, wenn die Handlung aufgehört hat (keine weiteren Aktionen in einem bestimmten Zeitraum). Die Zeitschwelle kann über brush.throttleDelay gesetzt werden.
- 'fixRate': Um das Event mit einer bestimmten Frequenz auszulösen. Die Frequenz kann über brush.throttleDelay gesetzt werden.

### 5.4.14 chart.brush.throttleDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

0 um Drosselung zu deaktivieren.

Standardmäßig wird brushSelected ausgelöst wann immer eine Auswahlbox ausgewählt oder bewegt wird, um die Umgebung über das Event zu informieren.

Zu häufiges Auslösen des Events kann jedoch zu Effizienzproblemen oder Beeinträchtigung der Animation führen. Deshalb stellt die brush-Komponente brush.throttleType und brush.throttleDelay zur Verfügung, um dieses Problem zu lösen.

Gültige Werte für throttleType sind:

- 'debounce': Um das Event erst auszulösen, wenn die Handlung aufgehört hat (keine weiteren Aktionen in einem bestimmten Zeitraum). Die Zeitschwelle kann über brush.throttleDelay gesetzt werden.
- 'fixRate': Um das Event mit einer bestimmten Frequenz auszulösen. Die Frequenz kann über brush.throttleDelay gesetzt werden.

### 5.4.15 chart.brush.removeOnClick

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Löschen aller Auswahlboxen beim Klick aktiviert ist wenn brush.brushMode auf 'single' steht.



### 5.4.16 chart.brush.inBrush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert visuelle Effekte für Elemente in der Auswahl.

Verfügbare visuelle Effekte sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Farbhelligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbwert in HSL.

In den meisten Fällen kann inBrush ungesetzt bleiben, wodurch die Standardkonfiguration verwendet wird.

### 5.4.17 chart.brush.outOfBrush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert visuelle Effekte für Elemente außerhalb der Auswahl.

Verfügbare visuelle Effekte sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Farbhelligkeit in HSL.

- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbwert in HSL.

**Hinweis:** Ist outOfBrush ungesetzt, so wird die Farbe standardmäßig auf '#ddd' gesetzt. Ist die Farbe nicht erwünscht kann man folgendes verwenden:

```
brush: {
  ...,
  outOfBrush: {
    colorAlpha: 0.1
  }
}
```

### 5.4.18 chart.brush.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10000 ]

z-Index der brush-Titelbox. Kann angepasst werden, wenn fehlerhafte Überlappungen auftreten.

## 5.5 ECharts (4) Calendar

### 5.5.1 chart.calendar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kalenderkoordinatensystem.

In ECharts können durch die Verwendung von Koordinatensystemen für die Erstellung des Kalenderdiagramms diese auch in Heatmaps, Streudiagrammen mit und ohne Effekte, oder Graphen verwendet werden.

### Kalenderlayout

Kalenderkoordinatensysteme können horizontal oder vertikal platziert werden. Von der Konvention her ist der Heatmap-Kalender horizontal. Benötigen wir in anderen Fällen aber eine höhere Zellgröße, kann die Gesamtbreite zu hoch sein. In diesem Fall hilft `calendar.orient`.

### Anpassung an Containergröße

Kalenderkoordinatensysteme können dazu konfiguriert werden, sich an die Containergröße anzupassen, was nützlich ist, wenn die Seitengröße unbekannt ist. Zum einen kann, wie bei anderen Komponenten, die Platzierung und Größe am Kalender festgelegt werden: `left` / `right` / `top` / `bottom` / `width` / `height`, welche es dem Kalender ermöglichen, seine Größe anhand der Containergröße zu modifizieren. Außerdem kann `cellSize` spezifiziert werden, um die Größe der Kalenderzellen festzulegen.

## 5.5.2 chart.calendar.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 5.5.3 chart.calendar.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

`zlevel`-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.5.4 chart.calendar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.5.5 chart.calendar.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Kalender-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 5.5.6 chart.calendar.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Kalender-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

### 5.5.7 chart.calendar.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Abstand zwischen der Kalender-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

### Erlaubte Werte

- {percent}
- {integer}
- auto

## 5.5.8 chart.calendar.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Kalender-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

### Erlaubte Werte

- {percent}
- {integer}

## 5.5.9 chart.calendar.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Breite des Kalenderkoordinatensystems.

Hinweis: cellSize ist standardmäßig 20. Ist width gesetzt, wird cellSize[0] auf 'auto' gesetzt

### 5.5.10 chart.calendar.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Höhe des Kalenderkoordinatensystems.

Hinweis: cellSize ist standardmäßig 20. Ist height gesetzt, wird cellSize[1] auf 'auto' gesetzt.

### 5.5.11 chart.calendar.range

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Benötigt, Spannweite des Kalenderkoordinatensystems, unterstützt mehrere Formate.

Beispiele:

```
// Ein Jahr
range: 2017

// Ein Monat
range: '2017-02'

// Eine Spanne
range: ['2017-01-02', '2017-02-23']

// Hinweis: Diese Spanne wird interpretiert als ['2017-01-01', '2017-02-01']
range: ['2017-01', '2017-02']
```

### 5.5.12 chart.calendar.cellSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Die Größe der Kalenderzellen. Kann als einzelner Wert oder als Array gesetzt werden, in dem das erste Element die Breite und das zweite die Höhe ist.

Unterstützt Selbstanpassung: 'auto', die standardmäßige Höhe und Breite ist 20.

Beispiele:

```
// Setze Breite und Höhe auf 20
cellSize: 20

// Setze Breite auf 20, und Höhe auf 40
cellSize: [20, 40]

// Setze Breite und Höhe auf 40
cellSize: [40]

// Setze Breite und Höhe auf Selbstanpassung
cellSize: 'auto'

// Setze Breite auf Selbstanpassung, und Höhe auf 40
cellSize: ['auto', 40]
```



### 5.5.13 chart.calendar.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Ausrichtung des Kalenders.

Erlaubte Werte

- horizontal
- vertical

### 5.5.14 chart.calendar.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinie des Kalenderkoordinatensystems.

### 5.5.15 chart.calendar.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen.

### 5.5.16 chart.calendar.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.5.17 chart.calendar.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Trennlinienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.5.18 chart.calendar.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Breite der Trennlinie.

## 5.5.19 chart.calendar.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.5.20 chart.calendar.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.5.21 chart.calendar.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.5.22 chart.calendar.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.5.23 chart.calendar.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.5.24 chart.calendar.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.5.25 chart.calendar.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil jedes Rechtecks im Kalenderkoordinatensystem.

### 5.5.26 chart.calendar.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Kalenderfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.5.27 chart.calendar.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: '#ccc' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.5.28 chart.calendar.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 5.5.29 chart.calendar.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 5.5.30 chart.calendar.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.5.31 chart.calendar.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.5.32 chart.calendar.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.5.33 chart.calendar.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.5.34 chart.calendar.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.5.35 chart.calendar.dayLabel.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Tage im Kalenderkoordinatensystem.

### 5.5.36 chart.calendar.dayLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tagesbeschriftungen angezeigt werden sollen.

### 5.5.37 chart.calendar.dayLabel.firstDay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Welcher Tag im Kalender als erstes aufgeführt wird, standardmäßig Sonntag.

Beispiel:

```
calendar: [{
  dayLabel: {
    firstDay: 1 // Starte mit Montag
  }
}]
```

### 5.5.38 chart.calendar.dayLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Abstand zwischen Tagesbeschriftung und Achsenlinie.

### 5.5.39 chart.calendar.dayLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position der Tagesbeschriftungen, am Anfang oder am Ende des Diagramms.

#### Erlaubte Werte

- start
- end

### 5.5.40 chart.calendar.dayLabel.nameMap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'en' ]

Bezeichnungen der Wochentage, standardmäßig 'en'. Unterstützt Chinesisch, Englisch und benutzerdefiniert. Index 0 ist immer Sonntag.

Beispiele:

```
// Abkürzung für Englisch ['S', 'M', 'T', 'W', 'T', 'F', 'S']
nameMap: 'en'

// Abkürzung für Chinesisch ['?', '?', '?', '?', '?', '?', '?']
nameMap: 'cn'

// Benutzerdefiniert: Englisch und Chinesisch gemischt, oder leer
nameMap: ['S', '?', 'T', '?', '?', '?', 'S']

calendar: [{
  dayLabel: {
    nameMap: 'en'
  }
}]
```

### 5.5.41 chart.calendar.dayLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Textfarbe.

### 5.5.42 chart.calendar.dayLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.5.43 chart.calendar.dayLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.5.44 chart.calendar.dayLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.5.45 chart.calendar.dayLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.5.46 chart.calendar.dayLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.5.47 chart.calendar.dayLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

### Erlaubte Werte

- top
- middle
- bottom

### 5.5.48 chart.calendar.dayLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.5.49 chart.calendar.dayLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	calendar

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.5.50 chart.calendar.dayLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.5.51 chart.calendar.dayLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.5.52 chart.calendar.dayLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.5.53 chart.calendar.dayLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.5.54 chart.calendar.dayLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.5.55 chart.calendar.dayLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.5.56 chart.calendar.dayLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.5.57 chart.calendar.dayLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.5.58 chart.calendar.dayLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.5.59 chart.calendar.dayLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.5.60 chart.calendar.dayLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.5.61 chart.calendar.dayLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.5.62 chart.calendar.dayLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.5.63 chart.calendar.dayLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.5.64 chart.calendar.dayLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.5.65 chart.calendar.dayLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.5.66 chart.calendar.dayLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.5.67 chart.calendar.monthLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Monatsbeschriftung im Kalenderkoordinatensystem.

### 5.5.68 chart.calendar.monthLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Monatsbeschriftungen angezeigt werden sollen.

### 5.5.69 chart.calendar.monthLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.5.70 chart.calendar.monthLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Der Abstand zwischen den Monatsbeschriftungen und der Achsenlinie.

### 5.5.71 chart.calendar.monthLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position der Monatsbeschriftungen, am Anfang oder am Ende des Diagramms.

Erlaubte Werte

- start
- end

### 5.5.72 chart.calendar.monthLabel.nameMap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'en' ]

Bezeichnungen der Monate, standardmäßig 'en'. Unterstützt Chinesisch, Englisch und benutzerdefiniert. Index 0 ist immer Januar.

Beispiele:

```
// Abkürzung für Englisch [
    'Jan', 'Feb', 'Mar',
    'Apr', 'May', 'Jun',
    'Jul', 'Aug', 'Sep',
    'Oct', 'Nov', 'Dec'
],
nameMap: 'en'

// Abkürzung für Chinesisch [
    '??', '??', '??',
    '??', '??', '??',
    '??', '??', '??',
    '??', '???'', '???'
]
nameMap: 'cn'

// Benutzerdefiniert: Englisch und Chinesisch gemischt oder leer
nameMap: [
    '??', 'Feb', '??',
    '??', 'May', '??',
    '??', '??', '',
    '??', 'Nov', '???'
],

calendar: [{
  monthLabel: {
    nameMap: 'en'
  }
}]
```

### 5.5.73 chart.calendar.monthLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Monatsbeschriftungen, unterstützt String-Templates und Callback-Funktionen.

Beispiele:

```
// Verwendung von String-Templates; z.B. 2017-02
/*
  Template-Variablen:
  {nameMap} standardmäßiges nameMap, z.B. 'Feb'
  {yyyy}   Vierstelliges Jahr, z.B. 2017
  {yy}    Zweistelliges Jahr, z.B. 17
  {MM}    Zweistelliger Monat, z.B. 02
  {M}     Einstelliger Monat, z.B. 2
*/
formatter: '{yyyy}-{MM}'

// Verwendung von Callback-Funktionen;
/*
  Funktionsparameter:
  param.nameMap standardmäßiges nameMap, z.B. 'Feb'
  param.yyyy   Vierstelliges Jahr, z.B. 2017
  param.yy    Zweistelliges Jahr, z.B. 17
  param.MM    Zweistelliger Monat, z.B. 02
  param.M     Einstelliger Monat, z.B. 2
*/
formatter: function (param) {
  // ...
  return param.MM;
}
```

### 5.5.74 chart.calendar.monthLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Textfarbe.

### 5.5.75 chart.calendar.monthLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.5.76 chart.calendar.monthLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.5.77 chart.calendar.monthLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.5.78 chart.calendar.monthLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.5.79 chart.calendar.monthLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.5.80 chart.calendar.monthLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.5.81 chart.calendar.monthLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.5.82 chart.calendar.monthLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.5.83 chart.calendar.monthLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.5.84 chart.calendar.monthLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.5.85 chart.calendar.monthLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.5.86 chart.calendar.monthLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.5.87 chart.calendar.monthLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.5.88 chart.calendar.monthLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.5.89 chart.calendar.monthLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.5.90 chart.calendar.monthLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.5.91 chart.calendar.monthLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.5.92 chart.calendar.monthLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.5.93 chart.calendar.monthLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.5.94 chart.calendar.monthLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.5.95 chart.calendar.monthLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.5.96 chart.calendar.monthLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.5.97 chart.calendar.monthLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.5.98 chart.calendar.monthLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.5.99 chart.calendar.yearLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Jahresbeschriftung im Kalenderkoordinatensystem.

### 5.5.100 chart.calendar.yearLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Jahresbeschriftungen angezeigt werden sollen.

### 5.5.101 chart.calendar.yearLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Der Abstand zwischen der Jahresbeschriftung und der Achsenlinie.

### 5.5.102 chart.calendar.yearLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Jahresbeschriftungen.

Standardwert: Wenn orient auf 'horizontal' steht, ist es 'left', wenn orient auf 'vertical' steht, ist es 'top'.

Erlaubte Werte

- top
- bottom
- left
- right

### 5.5.103 chart.calendar.yearLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Jahresbeschriftungen, unterstützt String-Templates und Callback-Funktionen.

Zeigt standardmäßig den Spann des momentanen Jahres an, oder, falls der Intervall über Jahre hinweg geht, das erste und das letzte Jahr.

Beispiele:

```
// Verwendung von String-Templates, z.B. ['2017-10-11', '2018-01-21']
/*
  Template-Variablen:
  {nameMap} Standardmäßige nameMap, z.B. '2017-2018'
  {start}   Startjahr, z.B. 2017
  {end}    Endjahr, z.B. 2018
*/
formatter: '{start}-{end}'

// Verwendung von Callback-Funktionen
/*
  Funktionsparameter:
  param.nameMap   Standardmäßige nameMap, z.B. '2017-2018'
  param.start    Startjahr, z.B. 2017
  param.end      Endjahr, z.B. 2018
*/
formatter: function (param) {
  // ...
  return param.end;
}
```

#### 5.5.104 chart.calendar.yearLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

#### 5.5.105 chart.calendar.yearLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.5.106 chart.calendar.yearLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.5.107 chart.calendar.yearLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.5.108 chart.calendar.yearLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.5.109 chart.calendar.yearLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.5.110 chart.calendar.yearLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.5.111 chart.calendar.yearLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.5.112 chart.calendar.yearLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:



```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.5.113 chart.calendar.yearLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.5.114 chart.calendar.yearLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.5.115 chart.calendar.yearLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.5.116 chart.calendar.yearLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.5.117 chart.calendar.yearLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.5.118 chart.calendar.yearLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.5.119 chart.calendar.yearLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.5.120 chart.calendar.yearLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.5.121 chart.calendar.yearLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.5.122 chart.calendar.yearLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.5.123 chart.calendar.yearLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 5.5.124 chart.calendar.yearLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 5.5.125 chart.calendar.yearLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.5.126 chart.calendar.yearLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.5.127 chart.calendar.yearLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.5.128 chart.calendar.yearLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.5.129 chart.calendar.yearLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.5.130 chart.calendar.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.



## 5.6 ECharts (4) Data

### 5.6.1 chart.data.hierachy.render

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: "default" ]

Wird für Sunburst-Diagramme verwendet, bei denen die Hierarchie der Daten festgelegt werden muss.

Zum Beispiel:

```
dashlet.addOption("chart.data.hierachy.render", "all");
```

#### Erlaubte Werte

- all

### 5.6.2 chart.dynamicSum.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Anzeige einer dynamischen Summe in einem Diagramm (Chart)

```
dashlet.addOption("chart.dynamicSum.show", "true");
```

### 5.6.3 chart.dynamicSum.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die Position des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.position", "top");
```

### 5.6.4 chart.dynamicSum.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die Farbe des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.color", "#000");
```

### 5.6.5 chart.dynamicSum.roundedDigits

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die roundedDigits des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.roundedDigits", "2");
```

### 5.6.6 chart.dynamicSum.excludeSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einige Serien von der dynamicSum-Berechnung in einem Diagramm auszuschließen

```
dashlet.addOption("chart.dynamicSum.excludeSeriesList","Motor City Art Classics;Unimax Art Galleris;Highway 66 Mini Classics");
```

### 5.6.7 chart.dynamicSum.retainStyleSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, den Stil einiger Serien beizubehalten, wenn die dynamicSum-Berechnung in einem Diagramm verwendet wird

```
dashlet.addOption("chart.dynamicSum.retainStyleSeriesList","Motor City Art Classics;Unimax Art Galleris");
```

### 5.6.8 chart.dynamicSum.retainStyleSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, den Stil einiger Serien beizubehalten, wenn die dynamicSum-Berechnung in einem Diagramm verwendet wird

```
dashlet.addOption("chart.dynamicSum.retainStyleSeriesList","Motor City Art Classics;Unimax Art Galleris");
```

### 5.6.9 chart.dynamicSum.customSeriesHandlerBlock

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einen benutzerdefinierten Code für die seriesHandler-Funktion einzufügen, um die dynamicSum in einem Diagramm zu berechnen

```
dashlet.addOption("chart.dynamicSum.customSeriesHandlerBlock","return null");
```

### 5.6.10 chart.dynamicSum.customHandleLegendSelectChangedBlock

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einen benutzerdefinierten Code für die LegendSelectChanged -Funktion einzufügen, um die dynamicSum in einem Diagramm zu berechnen

```
dashlet.addOption("chart.dynamicSum.customHandleLegendSelectChangedBlock", "return null");
```

## 5.7 ECharts (4) Data Zoom

### 5.7.1 chart.dataZoom.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die dataZoom-Komponente wird verwendet um auf einem spezifischen Bereich zu zoomen, wodurch Nutzer Daten genauer untersuchen, sich einen Überblick über die Daten verschaffen oder Ausreißer eliminieren können.

Diese Typen der dataZoom-Komponente werden unterstützt:

- dataZoomInside: Die Zoom-Funktionalität ist in das Koordinatensystem eingebettet, wodurch Nutzer durch Mausbewegungen oder Fingerberührungen (bei Touchscreens) das Koordinatensystem durchwandern und zoomen können.
- dataZoomSlider: Ein spezieller Schieberegler wird bereitgestellt, durch den Koordinatensysteme durch Mausbewegungen oder Fingerberührungen (bei Touchscreens) durchwandert oder gezoomt werden können.
- dataZoomSelect: Ein Marquee-Tool wird bereitgestellt, um das Koordinatensystem zu durchwandern oder zu zoomen. Dieses ist toolbox.feature.dataZoom, welches nur in toolbox konfiguriert werden kann.

## 5.7.2 chart.dataZoom.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Typ der dataZoom-Komponente.

Erlaubte Werte

- inside
- slider

## 5.7.3 chart.dataZoom.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 5.7.4 chart.dataZoom.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: true ]

Spezifiziert, ob die Komponente angezeigt werden soll. Steht der Wert auf false, wird sie nicht angezeigt, aber ihre Funktionen zum Filtern von Daten funktionieren trotzdem.

### 5.7.5 chart.dataZoom.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(47, 69, 84, 0)' ]

Die Hintergrundfarbe der Komponente.

### 5.7.6 chart.dataZoom.dataBackground.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil des Schattens der Daten

### 5.7.7 chart.dataZoom.dataBackground.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil des Schattens.

### 5.7.8 chart.dataZoom.dataBackground.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #2f4554 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.7.9 chart.dataZoom.dataBackground.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0.5 ]

Linienbreite.

## 5.7.10 chart.dataZoom.dataBackground.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.7.11 chart.dataZoom.dataBackground.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.7.12 chart.dataZoom.dataBackground.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.7.13 chart.dataZoom.dataBackground.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.7.14 chart.dataZoom.dataBackground.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.7.15 chart.dataZoom.dataBackground.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.7.16 chart.dataZoom.dataBackground.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Schattenbereichs.

### 5.7.17 chart.dataZoom.dataBackground.areaStyle.color

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(47,69,84,0.3) ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.7.18 chart.dataZoom.dataBackground.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.7.19 chart.dataZoom.dataBackground.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.7.20 chart.dataZoom.dataBackground.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.7.21 chart.dataZoom.dataBackground.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.7.22 chart.dataZoom.dataBackground.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.7.23 chart.dataZoom.fillerColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'rgba(47,69,84,0.25)' ]

Die Farbe, die den ausgewählten Bereich füllt.

### 5.7.24 chart.dataZoom.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ddd' ]

Die Farbe des Rahmens.

### 5.7.25 chart.dataZoom.handleIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'M8.2,13.6V3.9H6.3v9.7H3.1v14.9h3.3v9.7h1.8v-9.7h3.3V13.6H8.2z M9.7,24.4H4.8v-1.4h4.9V24.4z M9.7,19.1H4.8v-1.4h4.9V19.1z' ]

Bildliche Form des Handles, welche Pfadstrings unterstützt.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.7.26 chart.dataZoom.handleSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die Größe des Kontrollhandles. Kann in Pixeln angegeben werden, oder in Prozentwerten relativ zur Breite der dataZoom-Komponente. Ist standardmäßig gleich zur Breite der dataZoom-Komponente.

Erlaubte Werte

- {integer}
- {percent}

### 5.7.27 chart.dataZoom.handleStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Handles.

### 5.7.28 chart.dataZoom.handleStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #a7b7cc ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.7.29 chart.dataZoom.handleStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.7.30 chart.dataZoom.handleStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 5.7.31 chart.dataZoom.handleStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- solid
- dashed
- dotted

## 5.7.32 chart.dataZoom.handleStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 5.7.33 chart.dataZoom.handleStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.7.34 chart.dataZoom.handleStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.7.35 chart.dataZoom.handleStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.7.36 chart.dataZoom.handleStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.7.37 chart.dataZoom.labelPrecision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Präzision der Beschriftung wenn in Zahlenform. Wird standardmäßig anhand der Datenanzahl bestimmt.

### 5.7.38 chart.dataZoom.labelFormatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Das Formatierwerkzeug für die Beschriftung.

- Ist eine Template, falls ein String angegeben wird. Zum Beispiel 'aaaa{value}bbbb', wobei {value} durch den tatsächlichen Datenwert ersetzt wird.
- Ist eine Callback-Funktion, falls eine Funktion angegeben wird. Zum Beispiel:

```

/**
 * @param {*} value Wenn axis.type 'category' ist, ist `value` der Index von
axis.data.
 *                Sonst ist `value` der momentane Wert.
 * @param {string} valueStr Innerer formatierter String.
 * @return {string} Gibt die formatierte Beschriftung zurück.
 */
labelFormatter: function (value, valueStr) {
  return 'aaa' + value + 'bbb';
}

```

### 5.7.39 chart.dataZoom.showDetail

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Details angezeigt werden sollen. Das heißt, ob detaillierte Informationen über die Daten während dem Ziehen angezeigt werden sollen.

### 5.7.40 chart.dataZoom.showDataShadow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Spezifiziert, ob der Schatten der Daten in der dataZoom-Komponente angezeigt werden soll, wodurch die Tendenz der Daten schnell angezeigt werden kann.



### 5.7.41 chart.dataZoom.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht beim Ziehen aktualisiert werden soll. Steht der Wert auf false, wird die Ansicht erst aktualisiert, wenn nicht mehr gezogen wird.

### 5.7.42 chart.dataZoom.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil der dataZoom-Komponente.

### 5.7.43 chart.dataZoom.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

#### 5.7.44 chart.dataZoom.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

#### 5.7.45 chart.dataZoom.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.7.46 chart.dataZoom.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.7.47 chart.dataZoom.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.7.48 chart.dataZoom.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.7.49 chart.dataZoom.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.7.50 chart.dataZoom.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.7.51 chart.dataZoom.textStyle.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.7.52 chart.dataZoom.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.7.53 chart.dataZoom.textStyle.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.7.54 chart.dataZoom.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.7.55 chart.dataZoom.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.7.56 chart.dataZoom.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.7.57 chart.dataZoom.xAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche xAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Standardmäßig wird die erste xAxis kontrolliert, die parallel zum dataZoom liegt wenn dataZoom.orient auf 'horizontal' steht. Es wird aber empfohlen, den Wert explizit zu definieren.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```
option: {
  xAxis: [
    {...}, // Die erste xAxis
    {...}, // Die zweite xAxis
    {...}, // Die dritte xAxis
    {...} // Die vierte xAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      xAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                        // die erste und dritte xAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      xAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                  // die vierte xAxis kontrolliert
    }
  ]
}
```

### 5.7.58 chart.dataZoom.yAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche yAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Standardmäßig wird die erste yAxis kontrolliert, die parallel zum dataZoom liegt wenn dataZoom.orient auf 'vertical' steht. Es wird aber empfohlen, den Wert explizit zu definieren.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:



```

option: {
  yAxis: [
    {...}, // Die erste xAxis
    {...}, // Die zweite xAxis
    {...}, // Die dritte xAxis
    {...} // Die vierte xAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      yAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                        // die erste und dritte yAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      yAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                   // die vierte yAxis kontrolliert
    }
  ]
}

```

### 5.7.59 chart.dataZoom.radiusAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche radiusAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```

option: {
  radiusAxis: [
    {...}, // Die erste radiusAxis
    {...}, // Die zweite radiusAxis
    {...}, // Die dritte radiusAxis
    {...} // Die vierte radiusAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      radiusAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                             // die erste und dritte radiusAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      radiusAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                        // die vierte radiusAxis kontrolliert
    }
  ]
}

```

### 5.7.60 chart.dataZoom.angleAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche angleAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```

option: {
  angleAxis: [
    {...}, // Die erste angleAxis
    {...}, // Die zweite angleAxis
    {...}, // Die dritte angleAxis
    {...} // Die vierte angleAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      angleAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
        // die erste und dritte angleAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      angleAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
        // die vierte angleAxis kontrolliert
    }
  ]
}

```

### 5.7.61 chart.dataZoom.filterMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'filter' ]

Im Allgemeinen zoomt und bewegt sich die dataZoom-Komponente durch das Koordinatensystem, indem die Daten gefiltert und die Fenster der Achsen intern gesetzt werden.

Das Verhalten variiert je nach ausgewählten Filtereinstellungen (dataZoom.filterMode).

Mögliche Werte:

- 'filter': Daten außerhalb des Fensters werden **gefiltert**, was zu Veränderungen der Fenster auf anderen Achsen führen kann. Jedes Datenelement wird dann gefiltert, wenn eine seiner relevanten Dimensionen außerhalb des Fensters liegt.
- 'weakFilter': Daten außerhalb des Fensters werden **gefiltert**, was zu Veränderungen der Fenster auf anderen Achsen führen kann. Jedes Datenelement wird dann gefiltert, wenn alle seiner relevanten Dimensionen außerhalb des Fensters liegen.
- 'empty': Daten außerhalb des Fensters werden **auf NaN gesetzt**, wodurch sich die Fenster der anderen Achsen nicht verändern.
- 'none': Daten nicht filtern.

Wie filterMode gesetzt wird ist Sache des Nutzers, und hängt von den Anforderungen und Szenarien ab. Empirisch:

- Wenn nur die X- oder die Y-Achse von dataZoom kontrolliert wird, wird typischerweise filterMode: 'filter' verwendet, wodurch die anderen Achsen ihre Fenster automatisch an die gefilterten Daten anpassen können.
- Wenn sowohl die X- als auch die Y-Achse von dataZoom kontrolliert wird :
  - Wenn die Achsen sich nicht gegenseitig beeinflussen sollen (z.B. in einem Streudiagramm mit beiden Achsen auf Typ 'value'), sollten sie auf filterMode: 'empty' gesetzt werden.
  - Wenn die X-Achse die Hauptachse und die Y-Achse die Hilfsachse ist (oder umgekehrt) (z.B. in einem Balkendiagramm, wenn dataZoomX gezogen wird um das Fenster der X-Achse zu verändern, soll die Y-Achse sich an den neuen Datenbereich anpassen, wenn aber dataZoomY gezogen wird, soll die X-Achse sich nicht verändern), in diesem Fall sollte die X-Achse auf filterMode: 'filter' gesetzt werden, während die Y-Achse auf filterMode: 'empty' gesetzt werden sollte.

Dies wird im folgenden Beispiel demonstriert:

```
option = {
  dataZoom: [
    {
      id: 'dataZoomX',
      type: 'slider',
      xAxisIndex: [0],
      filterMode: 'filter'
    },
    {
      id: 'dataZoomY',
      type: 'slider',
      yAxisIndex: [0],
      filterMode: 'empty'
    }
  ],
  xAxis: {type: 'value'},
  yAxis: {type: 'value'},
  series{
    type: 'bar',
    data: [
      // Die erste Spalte gehört zur X-Achse,
      // die zweite Spalte gehört zur Y-Achse.
      [12, 24, 36],
      [90, 80, 70],
      [3, 9, 27],
      [1, 11, 111]
    ]
  }
}
```

Im oberen Beispiel steht dataZoomX auf 'filter'. Wenn der Nutzer dataZoomX zieht (und dataZoomY nicht verändert), und der Wertebereich dadurch auf [2, 50] verändert wird, geht dataZoomX durch die erste Spalte von series.data, und filtert alle Zeilen raus, deren Wert in der ersten Spalte nicht im Bereich [2, 50] liegt. Das Ergebnis für series.data ist:

```
[
  [12, 24, 36],
  // [90, 80, 70] Dieses Element wird gefiltert, da 90 nicht im Fenster ist.
  [3, 9, 27]
  // [1, 11, 111] Dieses Element wird gefiltert, da 1 nicht im Fenster ist.
]
```

Vor dem Filtern hatte die zweite Spalte, welche zur Y-Achse gehört, die Werte 24, 80, 9, 11. Nach dem Filtern sind nur 24 und 9 übrig. Das Ausmaß der Y-Achse ändert sich, um sich an die zwei Werte anzupassen (falls `yAxis.min` und `yAxis.max` nicht gesetzt sind).

Es kann also `filterMode: 'filter'` verwendet werden, damit sich die anderen Achsen an gefilterte Daten anpassen können.

Betrachten wir das Beispiel nun erneut. `dataZoomY` ist auf `filterMode: 'empty'` gesetzt. Wenn der Nutzer `dataZoomY` zieht (und `dataZoomX` nicht verändert), und der Wertebereich dadurch auf `[10, 60]` verändert wird, geht `dataZoomY` durch die zweite Spalte von `series.data`, und setzt alle Werte auf `NaN`, die nicht im Bereich `[10, 60]` liegen (`NaN` führt dazu, dass grafische Elemente, z.B. Balkenelemente, nicht angezeigt werden, aber weiter ihren Platz halten). Das Ergebnis für `series.data` ist:

```
[
  [12, 24, 36],
  [90, NaN, 70], // Auf NaN gesetzt
  [3, NaN, 27], // Auf NaN gesetzt
  [1, 11, 111]
]
```

In diesem Fall ändert sich die erste Spalte (also 12, 90, 3, 1, welche zur X-Achse gehört) nicht. Das Ziehen der Y-Achse ändert also nicht das Fenster der X-Achse, was nützlich für z.B. das Eliminieren von Ausreißern ist.

## Erlaubte Werte

- filter
- weakFilter
- empty
- none

## 5.7.62 chart.dataZoom.start

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der prozentuale Startwert des Fensters des Datenbereichs, in der Spanne von 0 bis 100. dataZoom.start und dataZoom.end definieren das Datenfenster in **prozentualer** Form.

### 5.7.63 chart.dataZoom.end

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Der prozentuale Endwert des Fensters des Datenbereichs, in der Spanne von 0 bis 100. dataZoom.start und dataZoom.end definieren das Datenfenster in **prozentualer** Form.

### 5.7.64 chart.dataZoom.startValue

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der absolute Startwert des Fensters, funktioniert nicht, wenn dataZoom.end gesetzt ist. dataZoom.startValue und dataZoom.endValue definieren das Datenfenster in **absoluter** Form.

Es ist anzumerken, dass, falls eine Achse als Kategorieachse gesetzt ist, startValue als Index des Arrays axis.data oder als Wert des Arrays gesetzt werden kann. Im letzteren Fall wird der Wert automatisch zu einem Index übersetzt.

### 5.7.65 chart.dataZoom.endValue

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der absolute Endwert des Fensters, funktioniert nicht, wenn dataZoom.end gesetzt ist.

dataZoom.startValue und dataZoom.endValue definieren das Datenfenster in **absoluter** Form.

Es ist anzumerken, dass, falls eine Achse als Kategorieachse gesetzt ist, endValue als Index des Arrays axis.data oder als Wert des Arrays gesetzt werden kann. Im letzteren Fall wird der Wert automatisch zu einem Index übersetzt.

### 5.7.66 chart.dataZoom.minSpan

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die minimale Fenstergröße, in Prozent, mit einem Wert im Bereich von [0, 100].

Wenn dataZoom.minValueSpan gesetzt ist, funktioniert minSpan nicht mehr.

### 5.7.67 chart.dataZoom.maxSpan

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Begrenzt die maximale Fenstergröße, in Prozent, mit einem Wert im Bereich von [0, 100].

Wenn dataZoom.maxValueSpan gesetzt ist, funktioniert maxSpan nicht mehr.

### 5.7.68 chart.dataZoom.minValueSpan

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die minimale Fenstergröße.

Zum Beispiel: Eine Zeitachse kann auf  $3600 * 24 * 1000 * 5$  gesetzt werden, um "5 Tage" darzustellen. Eine Kategorieachse kann auf 5 gesetzt werden, um 5 Kategorien darzustellen.

### 5.7.69 chart.dataZoom.maxValueSpan

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die maximale Fenstergröße.

Zum Beispiel: Eine Zeitachse kann auf  $3600 * 24 * 1000 * 5$  gesetzt werden, um "5 Tage" darzustellen. Eine Kategorieachse kann auf 5 gesetzt werden, um 5 Kategorien darzustellen.



### 5.7.70 chart.dataZoom.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob das Layout der dataZoom-Komponente horizontal oder vertikal ist. Bestimmt außerdem, ob im kartesischen Koordinatensystem standardmäßig die horizontale oder vertikale Achse kontrolliert werden soll.

Erlaubte Werte

- horizontal
- vertical

### 5.7.71 chart.dataZoom.zoomLock

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Größe des Fensters (des ausgewählten Bereichs) festgelegt sein soll.

Steht der Wert auf true, ist die Größe des Fensters festgelegt. Das heißt, nur die Translation (durch Ziehen mit Maus oder Berührung) steht zur Verfügung, das Zoomen nicht.

### 5.7.72 chart.dataZoom.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Spezifiziert die Rate der Aktualisierung der Ansicht, in millisekunden (ms).

Steht animation auf true und animationDurationUpdate auf einem Wert größer 0, kann throttle auf dem Standardwert 100 gelassen werden (oder auf einen Wert größer 0 gesetzt werden). Die Animation beim Ziehen ist sonst möglicherweise nicht flüssig.

Steht animation auf false oder animationDurationUpdate auf 0, und ist die Datenanzahl nicht sehr groß, so kann die Animation beim Ziehen unflüssig wirken. In diesem Fall hilft es, throttle auf 0 zu setzen.

### 5.7.73 chart.dataZoom.rangeMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zum Beispiel: rangeMode: ['value', 'percent'] bedeutet, dass absolute Werte am Anfang und prozentuale Werte am Ende verwendet werden sollen.

Erlaubte Werte

- value
- percent

### 5.7.74 chart.dataZoom.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.7.75 chart.dataZoom.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.7.76 chart.dataZoom.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 5.7.77 chart.dataZoom.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

### 5.7.78 chart.dataZoom.right

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}
- auto

### 5.7.79 chart.dataZoom.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}

### 5.7.80 chart.dataZoom.disabled

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der innere Zoom deaktiviert werden soll.

### 5.7.81 chart.dataZoom.zoomOnMouseWheel

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Wie das Zoomen ausgelöst wird. Mögliche Werte:

- true: Drehen des Mausekkrads löst Zoomen aus.
- false: Drehen des Mausekkrads löst kein Zoomen aus.
- 'shift': shift halten und Drehen des Mausekkrads löst Zoomen aus.
- 'ctrl': strg halten und Drehen des Mausekkrads löst Zoomen aus.
- 'alt': alt halten und Drehen des Mausekkrads löst Zoomen aus.

#### Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

## 5.7.82 chart.dataZoom.moveOnMouseMove

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Wie die Bewegung des Datenfensters ausgelöst wird. Mögliche Werte:

- true: Bewegung der Maus löst Bewegung des Datenfensters aus.
- false: Bewegung der Maus löst keine Bewegung des Datenfensters aus.
- 'shift': shift halten und Maus bewegen löst Bewegung des Datenfensters aus.
- 'ctrl': strg halten und Maus bewegen löst Bewegung des Datenfensters aus.
- 'alt': alt halten und Maus bewegen löst Bewegung des Datenfensters aus.

Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

## 5.7.83 chart.dataZoom.moveOnMouseWheel

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Wie die Bewegung des Datenfensters ausgelöst wird. Mögliche Werte:

- true: Drehen des Mousrads löst Bewegung des Datenfensters aus.
- false: Drehen des Mousrads löst keine Bewegung des Datenfensters aus.
- 'shift': shift halten und Drehen des Mousrads löst Bewegung des Datenfensters aus.
- 'ctrl': strg halten und Drehen des Mousrads löst Bewegung des Datenfensters aus.

- 'alt': alt halten und Drehen des Mausekkrads löst Bewegung des Datenfensters aus.

## Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

### 5.7.84 chart.dataZoom.preventDefaultMouseMove

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das standardmäßige Verhalten des 'Maus bewegt'-Events verhindert werden soll.

## 5.8 ECharts (4) Gantt

### 5.8.1 chart.gantt.column.categoryname

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-category

Legt die Spaltennummer für den CategoryName fest (Informationen werden auf der linken Grüne-Tab angezeigt). Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:



```
dashlet.addOption("chart.gantt.column.categoryname", "2");
```

## 5.8.2 chart.gantt.column.categorygroup

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-category

Legt die Spaltennummer für den CategoryGroup fest (Informationen werden neben der linken Grüne-Tab angezeigt). Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.categorygroup", "2");
```

## 5.8.3 chart.gantt.column.barinfo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt die Spaltennummer für die Informationen , die auf der Balken angezeigt wird. Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 0 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.barinfo", "2");
```

#### 5.8.4 chart.gantt.column.iconinfo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die Spaltennummer für die Icon Informationen an den linken Seiten des Gantt-charts. Diese Spalte ist benutzt für die Entscheidung ob die Icon angezeigt werden soll oder nicht. Diese Option ist mit die folgende Option benutzt: "chart.gantt.iconlink.check".

Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.iconinfo", "2");
```

#### 5.8.5 chart.gantt.column.barcolor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	bar-color, gantt

Legt die Spaltennummer für die Farbe der Balken. Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 0 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.barcolor", "2");
```

## 5.8.6 chart.gantt.screenitems

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-items

Legt fest, wie viele Datensätze im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.screenitems", "15");
```

## 5.8.7 chart.gantt.screenitems.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-items

Legt fest, die Höhe des Datensatzes im Gantt-Diagramm.

Beispiel:

```
dashlet.addOption("chart.gantt.screenitems.height", "15");
```

### 5.8.8 chart.gantt.iconlink.check

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt für die Entscheidung ob die Icon angezeigt werden soll oder nicht. Diese Option ist mit die folgende Option benutzt: "chart.gantt.column.iconinfo".

Beispiel:

```
dashlet.addOption("chart.gantt.iconlink.check", "'USA'");
```

### 5.8.9 chart.gantt.iconlink.path

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die SVG Image für das Icon fest.

Beispiel:

```
dashlet.addOption("chart.gantt.iconlink.path", "M369.9 97.9L286 14C277 5 264.8-.1
252.1-.1H48C21.5 0 0 21.5 0 48v416c0 26.5 21.5 48 48 48h288c26.5 0 48-21.5
48-48V131.9c0-12.7-5.1-25-14.1-34zM332.1 128H256V51.9l76.1 76.1zM48 464V48h160v104c0
13.3 10.7 24 24
24h104v288H48zm250.2-143.7c-12.2-12-47-8.7-64.4-6.5-17.2-10.5-28.7-25-36.8-46.3
3.9-16.1 10.1-40.6 5.4-56-4.2-26.2-37.8-23.6-42.6-5.9-4.4 16.1-.4 38.5 7 67.1-10
23.9-24.9 56-35.4 74.4-20 10.3-47 26.2-51 46.2-3.3 15.8 26 55.2 76.1-31.2 22.4-7.4
46.8-16.5 68.4-20.1 18.9 10.2 41 17 55.8 17 25.5 0 28-28.2 17.5-38.7zm-198.1
77.8c5.1-13.7 24.5-29.5 30.4-35-19 30.3-30.4 35.7-30.4 35zm81.6-190.6c7.4 0 6.7 32.1
1.8 40.8-4.4-13.9-4.3-40.8-1.8-40.8zm-24.4 136.6c9.7-16.9 18-37 24.7-54.7 8.3 15.1
18.9 27.2 30.1 35.5-20.8 4.3-38.9 13.1-54.8 19.2zm131.6-5s-5 6-37.3-7.8c35.1-2.6 40.9
5.4 37.3 7.8z");
```

### 5.8.10 chart.gantt.icon.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die Farbe für das Icon fest.

Beispiel:

```
dashlet.addOption("chart.gantt.icon.color", "#990000");
```

### 5.8.11 chart.gantt.nodata.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0

<b>Schlagworte / Tags</b>	gantt, nodatainformation
---------------------------	--------------------------

Legt den anzuzeigenden Text fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.text", "Keine Informationen");
```

### 5.8.12 chart.gantt.nodata.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text-Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.color", "#0000ff");
```

### 5.8.13 chart.gantt.nodata.textColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text-Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.textColor", "#0000ff");
```

### 5.8.14 chart.gantt.grid.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt die Höhe des Grid fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.height" , "'auto'");
```

### 5.8.15 chart.gantt.grid.left

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die linke Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.left" , "100");
```

### 5.8.16 chart.gantt.grid.top

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die obere Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.top" , "100");
```

### 5.8.17 chart.gantt.grid.right

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die rechte Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.right" , "100");
```

### 5.8.18 chart.gantt.grid.bottom

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die untere Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.bottom" , "100");
```

### 5.8.19 chart.gantt.nodata.maskColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.maskColor", "#0000ff");
```

### 5.8.20 chart.gantt.nodata.zlevel

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt die Z-Level für das Gantt-Diagramm.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.zlevel","0");
```

### 5.8.21 chart.gantt.nodata.fontSize

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text-Size fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.fontSize","12");
```

### 5.8.22 chart.gantt.nodata.showSpinner

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Wenn true festgelegt ist, wird der Spinner angezeigt, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.showSpinner","true");
```

### 5.8.23 chart.gantt.nodata.spinnerRadius

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den Radius des Spinners fest, der angezeigt wird, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.spinnerRadius","20");
```

### 5.8.24 chart.gantt.nodata.lineWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0

<b>Schlagworte / Tags</b>	gantt, nodatainformation
---------------------------	--------------------------

Legt den LineWidth des Spinners fest, der angezeigt wird, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.lineWidth","2");
```

### 5.8.25 chart.gantt.axis.path

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die SVG Image für das Axes des Gantt-Charts fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.path", "M0,0 L0,-20 L30,-20 C42,-20 38,-1 50,-1 L70,-1 L70,0 Z");
```

### 5.8.26 chart.gantt.axis.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2

<b>Schlagworte / Tags</b>	axis, gantt
---------------------------	-------------

Legt den X-Wert der linken oberen Ecke des Elements im Koordinatensystem der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.x", "0");
```

### 5.8.27 chart.gantt.axis.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den y-Wert der linken oberen Ecke des Elements im Koordinatensystem der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.y", "-20");
```

### 5.8.28 chart.gantt.axis.width

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Breite der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.width", "140");
```

### 5.8.29 chart.gantt.axis.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Höhe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.height", "40");
```

### 5.8.30 chart.gantt.axis.layout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt das Layout der Achse des Gantt-Diagramms fest. Mit dem Layout kann die Transformationsstrategie festgelegt werden.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.layout", "cover");
```

Erlaubte Werte

- cover
- center

### 5.8.31 chart.gantt.axis.style.fill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	

Die Farbe, die den Achsen-Pfad des Gantt-Diagramms ausfüllt.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.style.fill", "#368c6c");
```

### 5.8.32 chart.gantt.axis.categorynametext.style.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den x-Wert der linken oberen Ecke des Elements im Koordinatensystem für den Kategorienamen der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.x", "4");
```

### 5.8.33 chart.gantt.axis.categorynametext.style.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den y-Wert der linken oberen Ecke des Elements im Koordinatensystem für den Kategorienamen der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.y", "-3");
```

### 5.8.34 chart.gantt.axis.categorynametext.style.textVerticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die vertikale Textausrichtung für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:



```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textVerticalAlign",
"bottom");
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.8.35 chart.gantt.axis.categorynametext.style.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die horizontale Textausrichtung für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textAlign", "left");
```

#### Erlaubte Werte

- left
- center
- right

### 5.8.36 chart.gantt.axis.categorynametext.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Textfarbe für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textFill", "#fff");
```

### 5.8.37 chart.gantt.axis.categorygroupertext.style.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den X-Wert der linken oberen Ecke des Elements im Koordinatensystem für die Kategoriegruppe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.x", "125");
```

### 5.8.38 chart.gantt.axis.categorygroupertext.style.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagnworte / Tags</b>	axis, gantt

Legt den Y-Wert der linken oberen Ecke des Elements im Koordinatensystem für die Kategoriegruppe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.y", "-2");
```

### 5.8.39 chart.gantt.axis.categorygroupertext.style.textVerticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagnworte / Tags</b>	axis, gantt

Legt die vertikale Textausrichtung für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.textVerticalAlign",  
"bottom");
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.8.40 chart.gantt.axis.categorygroup.text.style.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die horizontale Textausrichtung für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroup.text.style.textAlign", "left");
```

Erlaubte Werte

- left
- center
- right

### 5.8.41 chart.gantt.axis.categorygroup.text.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Text-Farbe für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroup.text.style.textFill", "#fff");
```

### 5.8.42 chart.gantt.rect.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Text-Farbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.textFill", "#000");
```

### 5.8.43 chart.gantt.rect.style.stroke

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Stroke-Farbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.stroke", "transparent");
```

#### 5.8.44 chart.gantt.rect.style.fill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Füllfarbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.fill", "transparent");
```

## 5.9 ECharts (4) General

### 5.9.1 chart.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	map

Setzt der Typ des Charts.

Beispiel:

```
dashlet.addOption("chart.type", "bar");
```

## 5.9.2 chart.theme

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	theme

Legt die Thema für das Dashlet fest.

Beispiel:

```
<span style="color:black; font-size:10.0pt">dashlet.addOption</span>("chart.theme", "infographics");
```

## 5.10 ECharts (4) Geo

### 5.10.1 chart.geo.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Komponente für geografische Koordinatensysteme.

Diese Komponente wird verwendet um Karten zu zeichnen, Streuserien und Balkenserien werden unterstützt.

Ab 3.1.10 unterstützt die geo-Komponente auch Maus-Events, deren Parameter sind:

```

{
  componentType: 'geo',
  // Index der geo-Komponente in den Optionen
  geoIndex: number,
  // Name des geklickten Gebiets, z.B. Shanghai
  name: string,
  // Geklicktes Regionsobjekt als Input, siehe geo.regions
  region: Object
}

```

**Tip:** Die Regionsfarbe kann auch über Kartenserien gesteuert werden. Siehe `series-map.geoIndex`.

### 5.10.2 chart.geo.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.10.3 chart.geo.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die geo-Komponente angezeigt werden soll.



## 5.10.4 chart.geo.map

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '' ]

Kartendiagramme.

Wegen der zunehmenden Feinheit von Karten enthält ECharts 3 aus Gründen der Paketgröße standardmäßig keine Kartendaten. Benötigte Kartendateien können auf der Karten-Downloadseite gefunden und dann in ECharts eingebunden und registriert werden.

ECharts stellt zwei Formate für Kartendaten zur Verfügung. Das eine kann in <script>-Tags als JavaScript-Datei eingebunden werden, das andere ist in JSON-Format und sollte mit AJAX geladen werden. Sobald die JavaScript-Datei geladen ist, werden Kartename und -daten automatisch geladen, während der Name in der JSON-Form explizit zugewiesen werden muss.

Hier sind Beispiele der beiden Typen:

### JavaScript Import Beispiel

```
<script src="echarts.js"></script>
<script src="map/js/china.js"></script>
<script>
var chart = echarts.init(document.getElementById('main'));
chart.setOption({
  series: [{
    type: 'map',
    map: 'china'
  }]
});
</script>
```

### JSON Import Beispiel

```
$.get('map/json/china.json', function (chinaJson) {
  echarts.registerMap('china', chinaJson);
  var chart = echarts.init(document.getElementById('main'));
  chart.setOption({
    series: [{
      type: 'map',
      map: 'china'
    }]
  });
});
```

ECharts verwendet das geoJSON-Format für Kartenumrandungen. Neben den oben vorgestellten Methoden kann man, wenn man will, geoJSON-Daten auch mit anderen Methoden beschaffen und in ECharts registrieren.

### 5.10.5 chart.geo.roam

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Zoomen und Bewegen mit der Maus aktiviert werden soll, standardmäßig false. Wenn entweder zoomen oder bewegen gewünscht ist, kann der Wert auf 'scale' oder 'move' gesetzt werden. Andernfalls kann er auf true gesetzt werden, um beides zu aktivieren.

#### Erlaubte Werte

- {boolean}
- scale
- move

### 5.10.6 chart.geo.center

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zentrum des momentanen Viewports, in Längen- und Breitengraden.

Beispiel:

```
center: [115.97, 29.71]
```

### 5.10.7 chart.geo.aspectScale

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.75 ]

Wird verwendet, um das Seitenverhältnis von geo zu skalieren.

Das endültige Seitenverhältnis berechnet sich aus:  $\text{geoBoundingRect.width} / \text{geoBoundingRect.height} * \text{aspectScale}$ .

### 5.10.8 chart.geo.boundingCoords

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zweidimensionales Array. Definiert die Koordinaten von oben-links und unten-rechts in der Layout-Box

```
// Eine komplette Weltkarte
map: 'world',
left: 0, top: 0, right: 0, bottom: 0,
boundingCoords: [
  // [Längengrad, Breitengrad] der Ecke oben-links
  [-180, 90],
  // [Längengrad, Breitengrad] der Ecke unten-rechts
  [180, -90]
],
```

### 5.10.9 chart.geo.zoom

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Zoom-Rate des momentanen Viewports.

### 5.10.10 chart.geo.scaleLimit.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grenzen der Skalierung, mit min und max. Standardmäßig 1.

### 5.10.11 chart.geo.scaleLimit.min

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimale Skalierung.

### 5.10.12 chart.geo.scaleLimit.max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Maximale Skalierung

### 5.10.13 chart.geo.nameMap

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mapping der Namen für angepasste Gebiete. Zum Beispiel:

```
{
  'China' : '??'
}
```

### 5.10.14 chart.geo.selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Entscheidung, ob mehrfache Auswahl unterstützt wird. Standardmäßig steht der Wert auf false um die Auswahl zu deaktivieren. Der Wert kann auch 'single' für das Auswählen einzelner Gebiete, oder 'multiple' für das Auswählen mehrerer Gebiete sein.

### 5.10.15 chart.geo.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 5.10.16 chart.geo.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.10.17 chart.geo.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

#### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

#### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight

- {integer}
- {percent}

### 5.10.18 chart.geo.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.10.19 chart.geo.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.10.20 chart.geo.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 5.10.21 chart.geo.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.10.22 chart.geo.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.23 chart.geo.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.10.24 chart.geo.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

#### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.10.25 chart.geo.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.10.26 chart.geo.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.10.27 chart.geo.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

**Erlaubte Werte**

- left
- center
- right

**5.10.28 chart.geo.label.verticalAlign**

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.10.29 chart.geo.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.10.30 chart.geo.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.31 chart.geo.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.32 chart.geo.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.10.33 chart.geo.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.10.34 chart.geo.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.10.35 chart.geo.label.shadowColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.10.36 chart.geo.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.10.37 chart.geo.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.10.38 chart.geo.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.10.39 chart.geo.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.40 chart.geo.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.10.41 chart.geo.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 5.10.42 chart.geo.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.10.43 chart.geo.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.10.44 chart.geo.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.10.45 chart.geo.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.10.46 chart.geo.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.10.47 chart.geo.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.10.48 chart.geo.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 5.10.49 chart.geo.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.10.50 chart.geo.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

**Erlaubte Werte**

- top
- left
- right

- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight

### 5.10.51 chart.geo.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.10.52 chart.geo.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.10.53 chart.geo.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.10.54 chart.geo.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.10.55 chart.geo.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.56 chart.geo.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.10.57 chart.geo.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

#### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.10.58 chart.geo.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.10.59 chart.geo.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.10.60 chart.geo.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

**Erlaubte Werte**

- left
- center
- right

**5.10.61 chart.geo.label.emphasis.verticalAlign**

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.10.62 chart.geo.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.10.63 chart.geo.label.emphasis.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 5.10.64 chart.geo.label.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 5.10.65 chart.geo.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.10.66 chart.geo.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.10.67 chart.geo.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.10.68 chart.geo.label.emphasis.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.10.69 chart.geo.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.10.70 chart.geo.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.10.71 chart.geo.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.10.72 chart.geo.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.73 chart.geo.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.10.74 chart.geo.label.emphasis.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 5.10.75 chart.geo.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.10.76 chart.geo.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.10.77 chart.geo.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.10.78 chart.geo.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.10.79 chart.geo.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.10.80 chart.geo.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.10.81 chart.geo.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grafikstil des Randes eines Kartengebiets, emphasis ist der Stil im hervorgehobenen Zustand, also wenn die Maus auf den Bereich zeigt oder er über die Legendenkomponente hervorgehoben wird.

### 5.10.82 chart.geo.itemStyle.areaColor.object

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Den Bereich füllende Farbe.

### 5.10.83 chart.geo.itemStyle.areaColor.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.10.84 chart.geo.itemStyle.areaColor.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.10.85 chart.geo.itemStyle.areaColor.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 5.10.86 chart.geo.itemStyle.areaColor.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 5.10.87 chart.geo.itemStyle.areaColor.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.10.88 chart.geo.itemStyle.areaColor.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.10.89 chart.geo.itemStyle.areaColor.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.10.90 chart.geo.itemStyle.areaColor.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.10.91 chart.geo.itemStyle.areaColor.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.10.92 chart.geo.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil von Kartenbereich im hervorgehobenen Zustand.

### 5.10.93 chart.geo.itemStyle.emphasis.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Den Bereich füllende Farbe.

### 5.10.94 chart.geo.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.10.95 chart.geo.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.10.96 chart.geo.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.10.97 chart.geo.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.10.98 chart.geo.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.10.99 chart.geo.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.10.100 chart.geo.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.10.101 chart.geo.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.10.102 chart.geo.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.10.103 chart.geo.zlevel

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

#### 5.10.104 chart.geo.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

#### 5.10.105 chart.geo.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

#### 5.10.106 chart.geo.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle

- bottom

### 5.10.107 chart.geo.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 5.10.108 chart.geo.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

## Erlaubte Werte

- {percent}
- {integer}

## 5.10.109 chart.geo.layoutCenter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

layoutCenter und layoutSize stellen eine andere Layout-Strategie als left/right/top/bottom/width/height zur Verfügung.

Bei der Benutzung von left/right/top/bottom/width/height ist es schwierig, die Karte in einer Box mit festem Breite-Höhe-Verhältnis zu platzieren. In diesem Fall kann das layoutCenter-Attribut benutzt werden, um die Position des Zentrums der Karte zu definieren, und layoutSize kann benutzt werden, um die Größe der Karte zu definieren. Zum Beispiel:

```
layoutCenter: ['30%', '30%'],
// Ist das Breite-Höhe-Verhältnis größer als 1, wird die Breite auf 100 gesetzt.
// Sonst wird die Höhe auf 100 gesetzt.
// So wird sichergestellt, dass die Karten den 100x100 Bereich nicht überschreitet.
layoutSize: 100
```

Wurden diese beiden Werte gesetzt, so sind left/right/top/bottom/width/height ungültig.

## 5.10.110 chart.geo.layoutSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Karte, siehe layoutCenter für weitere Informationen. Prozentuale Werte relativ zur Bildschirmbreite und absolute Pixelwerte werden unterstützt.

### 5.10.111 chart.geo.regions.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil für bestimmte Regionen konfigurieren. Zum Beispiel:

```
regions: [{
  name: 'Guangdong',
  itemStyle: {
    areaColor: 'red',
    color: 'red'
  }
}]
```

Die Regionsfarbe kann auch über Kartenserien kontrolliert werden. Siehe series-map.geolIndex

### 5.10.112 chart.geo.regions.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name eines Gebiets auf der Karte, wie 'Guangdong' oder 'Zhejiang'.

### 5.10.113 chart.geo.regions.{x}.selected

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob dieses Gebiet ausgewählt ist.

#### 5.10.114 chart.geo.regions.{x}.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil dieses Bereichs.

#### 5.10.115 chart.geo.regions.{x}.itemStyle.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsfarbe in der Karte

### 5.10.116 chart.geo.regions.{x}.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.10.117 chart.geo.regions.{x}.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.10.118 chart.geo.regions.{x}.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 5.10.119 chart.geo.regions.{x}.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 5.10.120 chart.geo.regions.{x}.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.10.121 chart.geo.regions.{x}.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.10.122 chart.geo.regions.{x}.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.10.123 chart.geo.regions.{x}.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.10.124 chart.geo.regions.{x}.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.10.125 chart.geo.regions.{x}.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil eines hervorgehobenen BereichsBereichs.

### 5.10.126 chart.geo.regions.{x}.itemStyle.emphasis.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsfarbe in der Karte

### 5.10.127 chart.geo.regions.{x}.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.10.128 chart.geo.regions.{x}.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.10.129 chart.geo.regions.{x}.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.10.130 chart.geo.regions.{x}.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 5.10.131 chart.geo.regions.{x}.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.10.132 chart.geo.regions.{x}.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.10.133 chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.10.134 chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 5.10.135 chart.geo.regions.{x}.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.10.136 chart.geo.regions.{x}.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 5.10.137 chart.geo.regions.{x}.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.10.138 chart.geo.regions.{x}.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]

Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.10.139 chart.geo.regions.{x}.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.10.140 chart.geo.regions.{x}.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

#### 5.10.141 chart.geo.regions.{x}.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

#### 5.10.142 chart.geo.regions.{x}.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

##### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 5.10.143 chart.geo.regions.{x}.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.144 chart.geo.regions.{x}.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.10.145 chart.geo.regions.{x}.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bolder
- bold
- lighter
- {integer}

### 5.10.146 chart.geo.regions.{x}.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.10.147 chart.geo.regions.{x}.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.10.148 chart.geo.regions.{x}.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.10.149 chart.geo.regions.{x}.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.10.150 chart.geo.regions.{x}.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.10.151 chart.geo.regions.{x}.label.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.152 chart.geo.regions.{x}.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.153 chart.geo.regions.{x}.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.10.154 chart.geo.regions.{x}.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.10.155 chart.geo.regions.{x}.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.10.156 chart.geo.regions.{x}.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.10.157 chart.geo.regions.{x}.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.10.158 chart.geo.regions.{x}.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.10.159 chart.geo.regions.{x}.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.10.160 chart.geo.regions.{x}.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.161 chart.geo.regions.{x}.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.162 chart.geo.regions.{x}.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.163 chart.geo.regions.{x}.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.10.164 chart.geo.regions.{x}.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.10.165 chart.geo.regions.{x}.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.10.166 chart.geo.regions.{x}.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.10.167 chart.geo.regions.{x}.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.10.168 chart.geo.regions.{x}.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.10.169 chart.geo.regions.{x}.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts



3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 5.10.170 chart.geo.regions.{x}.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.10.171 chart.geo.regions.{x}.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight

### 5.10.172 chart.geo.regions.{x}.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.10.173 chart.geo.regions.{x}.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.10.174 chart.geo.regions.{x}.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.10.175 chart.geo.regions.{x}.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.10.176 chart.geo.regions.{x}.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.177 chart.geo.regions.{x}.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.10.178 chart.geo.regions.{x}.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.10.179 chart.geo.regions.{x}.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.10.180 chart.geo.regions.{x}.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 5.10.181 chart.geo.regions.{x}.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 5.10.182 chart.geo.regions.{x}.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 5.10.183 chart.geo.regions.{x}.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 5.10.184 chart.geo.regions.{x}.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.



Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.185 chart.geo.regions.{x}.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.186 chart.geo.regions.{x}.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.10.187 chart.geo.regions.{x}.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.10.188 chart.geo.regions.{x}.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.10.189 chart.geo.regions.{x}.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.10.190 chart.geo.regions.{x}.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.10.191 chart.geo.regions.{x}.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.10.192 chart.geo.regions.{x}.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.10.193 chart.geo.regions.{x}.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.194 chart.geo.regions.{x}.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.10.195 chart.geo.regions.{x}.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.10.196 chart.geo.regions.{x}.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.10.197 chart.geo.regions.{x}.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.10.198 chart.geo.regions.{x}.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.10.199 chart.geo.regions.{x}.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.10.200 chart.geo.regions.{x}.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.10.201 chart.geo.regions.{x}.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.10.202 chart.geo.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.



## 5.11 ECharts (4) Graphic

### 5.11.1 chart.graphic.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Graphic
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Grafik-Komponente ermöglicht das Erstellen von grafischen Elementen in ECharts.

Die folgenden Grafiktypen werden unterstützt:

image, text, circle, sector, ring, polygon, polyline, rect, line, bezierCurve, arc, group

#### **Konfiguration der grafischen Komponente**

Eine einfache Art, um ein grafisches Element zu definieren:

```
myChart.setOption({
  ...,
  graphic: {
    type: 'image',
    ...
  }
});
```

Mehrere grafische Elemente definieren:

```
myChart.setOption({
  ...,
  graphic: [
    { // Ein 'image'-Element.
      type: 'image',
      ...
    },
    { // Ein 'text'-Element mit spezifizierter id.
      type: 'text',
      id: 'text1',
      ...
    },
    { // Ein 'group'-Element, in dem Kinder definiert werden können.
      type: 'group',
      children: [
        {
          type: 'rect',
          id: 'rect1',
          ...
        },
        {
          type: 'image',
          ...
        },
        ...
      ]
    },
    ...
  ]
});
```

Ersetzen und entfernen von Elementen mit setOption:

```
myChart.setOption({
  ...,
  graphic: [
    { // Das oben definierte Element 'text1' entfernen.
      id: 'text1',
      $action: 'remove',
      ...
    },
    { // Das oben definierte 'rect1' durch einen Zirkel ersetzen.
      // Man merke, obwohl 'rect1' ein Kind einer Gruppe ist,
      // ist es nicht nötig die Ebenenbeziehung erneut zu beachten,
      // wenn id benutzt wird, um das Element zu spezifizieren.
      id: 'rect1',
      $action: 'replace',
      type: 'circle',
      ...
    }
  ]
});
```

Es ist anzumerken, dass bei der Modifizierung von existierenden Elementen mit `setOption`, falls `id` nicht spezifiziert ist, die neuen Optionen auf die existierenden Elemente anhand ihrer Reihenfolge gemappt werden, was unerwartete Ergebnisse haben kann. Daher wird normalerweise die Verwendung von `id` empfohlen.

### **Konfiguration von grafischen Elementen**

Verschiedene Typen von grafischen Elementen haben ihre eigenen jeweiligen Konfigurationen, aber die folgenden Attribute haben sie gemein:

```
{
  // id wird verwendet, um das Element bei Updates zu spezifizieren.
  // id kann ignoriert werden, falls sie nicht benötigt wird.
  id: 'xxx',

  // Spezifiziert Elementtyp. Kann bei der ersten Definition eines Elements nicht
  ignoriert werden.
  type: 'image',

  // Alle folgenden Attribute können ignoriert werden und kriegen dann
  Standardwerte zugewiesen.

  // Spezifiziert die Operation, die auf dem Element ausgeführt werden soll, wenn
  `setOption` aufgerufen wird.
  // Standardwert ist 'merge', auch möglich ist 'replace' und 'remove'.
  $action: 'replace',

  // Diese vier Eigenschaften werden verwendet, um das Element zu platzieren.
  // Jeder Wert kann absolut (wie 10, was 10 Pixel bedeutet), relativ (wie '12%')
  oder 'center'/'middle' sein.
  left: 10,
  // right: 10,
  top: 'center',
  // bottom: '10%',

  shape: {
    // Hier sind Einstellungen für die Form, wie `x`, `y`, `cx`, `cy`, `width`,
    // `height`, `r`, `points`, ...
    // Man merke, falls `left`/`right`/`top`/`bottom` gesetzt wurden,
    funktionieren `x`/`y`/`cx`/`cy` hier nicht.
  },

  style: {
    // Hier sind Einstellungen für den Stil des Elements, wie `fill`, `stroke`,
    // `lineWidth`, `shadowBlur`, ...
  },

  // z-Wert der Elemente.
  z: 10,
  // Ob auf Maus-Events / Touch-Events reagiert werden soll.
  silent: true,
  // Ob das Element unsichtbar ist.
  invisible: false,
  // Spezifiziert, ob das gesamte transformierte Element (Kinder beinhaltend, falls
  es group ist)
  // durch seinen Container begrenzt wird. Mögliche Werte sind 'raw', 'all'.
  bounding: 'raw',
  // Kann gezogen werden oder nicht.
  draggable: false,
  // Event-handler, kann auch onmousemove, ondrag, ... sein (unten aufgeführt)
  onclick: function () {...}
}
```

## Event-handlers von grafischen Elementen

Diese Events werden unterstützt: onclick, onmouseout, onmouseover, onmousewheel, onmousedown, onmouseup, ondrag, ondragstart, ondragend, ondragcenter, ondragleave, ondragover, ondrop.

## Hierarchie von grafischen Elementen

Nur group-Elemente haben Kinder, wodurch eine Gruppe von Elementen zusammen positioniert und transformiert werden kann.

## Konfiguration der Form grafischer Elemente

Elemente mit verschiedenen Typen haben entsprechend unterschiedliche Einstellungen ihrer Form. Zum Beispiel:

```
{
  type: 'rect',
  shape: {
    x: 10,
    y: 10,
    width: 100,
    height: 200
  }
},
{
  type: 'circle',
  shape: {
    cx: 20,
    cy: 30,
    r: 100
  }
},
{
  type: 'image',
  style: {
    image: 'http://xxx.xxx.xxx/a.png',
    x: 100,
    y: 200,
    width: 230,
    height: 400
  }
},
{
  type: 'text',
  style: {
    text: 'Dieser text',
    x: 100,
    y: 200
  }
}
```

## Transformation und absolute Positionierung grafischer Elemente

Elemente können transformiert werden (Translation, Rotation, Skalierung). Siehe position, rotation, scale, origin

Zum Beispiel:

```
{
  type: 'rect', // oder beliebiger anderer Typ.

  // Translation, verwendet standardmäßig [0, 0].
  position: [100, 200],

  // Skalierung, verwendet standardmäßig [1, 1].
  scale: [2, 4],

  // Rotation, verwendet standardmäßig 0. Negative Werte stehen für Rotation im
  // Uhrzeigersinn.
  rotation: Math.PI / 4,

  // Ursprungspunkt der Rotation und Skalierung, verwendet standardmäßig [0, 0].
  origin: [10, 20],

  shape: {
    // ...
  }
}
```

- Jedes Element wird im Koordinatensystem seines Elternteils transformiert, insbesondere kann die Transformation von einem Element und seinem Elternteil "gestapelt" werden.
- Die Transformation wird in dieser Reihenfolge durchgeführt:
  - a. Translation [-el.origin[0], -el.origin[1]].
  - b. Skalieren nach el.scale.
  - c. Rotation nach el.rotation.
  - d. Rücktranslation nach el.origin.
  - e. Translation nach el.position.
- Namentlich, Skalierung und Rotation zuerst, dann Translation. Durch diesen Vorgang beeinträchtigt die Translation nicht den Ursprung der Skalierung und Rotation.

### Relative Positionierung

In echten Anwendungen ist die Größe eines Containers nicht immer fest. Eine Möglichkeit zur relativen Positionierung wird also benötigt. In der Grafik-Komponente werden left / right / top / bottom / width / height verwendet, um Elemente relativ zu positionieren.

Zum Beispiel:

```

{ // Positioniere das Bild an der unteren Mitte des Containers.
  type: 'image',
  left: 'center', // An der horizontalen Mitte positionieren.
  bottom: '10%', // 10% über der unteren Grenze positionieren.
  style: {
    image: 'http://xxx.xxx.xxx/a.png',
    width: 45,
    height: 45
  }
},
{ // Die gesamte Gruppe in der rechten unteren Ecke des Containers positionieren.
  type: 'group',
  right: 0, // Positionierung an der rechten Grenze.
  bottom: 0, // Positionierung an der unteren Grenze.
  rotation: Math.PI / 4,
  children: [
    {
      type: 'rect',
      left: 'center', // Positionierung an der horizontalen Mitte nach dem
      Elternteil.
      top: 'middle', // Positionierung an der vertikalen Mitte nach dem
      Elternteil.
      shape: {
        width: 190,
        height: 90
      },
      style: {
        fill: '#fff',
        stroke: '#999',
        lineWidth: 2,
        shadowBlur: 8,
        shadowOffsetX: 3,
        shadowOffsetY: 3,
        shadowColor: 'rgba(0,0,0,0.3)'
      }
    },
    {
      type: 'text',
      left: 'center', // Positionierung an der horizontalen Mitte nach dem
      Elternteil.
      top: 'middle', // Positionierung an der vertikalen Mitte nach dem
      Elternteil.
      style: {
        fill: '#777',
        text: [
          'Dies ist Text',
          'Dies ist Text',
          'Text ausgeben'
        ].join('\n'),
        font: '14px Microsoft YaHei'
      }
    }
  ]
}

```

```
    ]
  }
```

Es ist anzumerken, dass `bounding` verwendet werden kann, um zu spezifizieren, ob das gesamte transformierte Element (Kinder beinhaltend, falls es `group` ist) durch seinen Container begrenzt wird.

## 5.12 ECharts (4) Grid

### 5.12.1 `chart.grid.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeichengitter in einem rechteckigen Koordinatensystem. In einem einzelnen Gitter sind maximal zwei X- und Y-Achsen erlaubt. Liniendiagramme, Balkendiagramme und Streudiagramme (Blasendiagramme) können in Gittern gezeichnet werden.

In ECharts 2.x konnte es nur maximal eine Gitterkomponente in einer einzelnen ECharts Instanz geben. In ECharts 3 gibt es jedoch keine solche Einschränkung.

### 5.12.2 `chart.grid.{x}.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.



### 5.12.3 chart.grid.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Gitter im rechteckigen Koordinatensystem angezeigt werden soll.

### 5.12.4 chart.grid.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.12.5 chart.grid.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.12.6 chart.grid.{x}.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '10%' ]

Abstand zwischen der Gitterkomponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 5.12.7 chart.grid.{x}.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Gitter-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.12.8 chart.grid.{x}.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '10%' ]

Abstand zwischen der Gitterkomponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

#### Erlaubte Werte

- {integer}
- {percent}

### 5.12.9 chart.grid.{x}.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Gitterkomponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Erlaubte Werte

- {integer}
- {percent}

### 5.12.10 chart.grid.{x}.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Gitterkomponente. Standardmäßig adaptiv.

### 5.12.11 chart.grid.{x}.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Gitterkomponente. Standardmäßig adaptiv.

### 5.12.12 chart.grid.{x}.containLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Gitterregion die Achsentick-Beschriftungen der Achsen enthält.

- Wenn containLabel false ist:
  - grid.left grid.right grid.top grid.bottom grid.width grid.height bestimmen die Platzierung und Größe des Rechtecks, das aus xAxis und yAxis besteht.
  - Auf false setzen hilft, wenn mehrere Gitter anhand ihrer Achsen ausgerichtet werden müssen.
- Wenn containLabel true ist:
  - grid.left grid.right grid.top grid.bottom grid.width grid.height bestimmen die Platzierung und Größe des Rechtecks, das nicht nur die Achsen, sondern auch die Beschriftungen dieser Achsen enthält.
  - Auf true setzen hilft, wenn die länge der Achsenbeschriftungen dynamisch und schwer abzuschätzen ist, um zu verhindern, dass diese aus dem Container laufen oder andere Elemente überdecken

### 5.12.13 chart.grid.{x}.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Gitters, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 5.12.14 chart.grid.{x}.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Gitters. Unterstützt das gleiche Farbformat wie backgroundColor.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 5.12.15 chart.grid.{x}.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Gitters.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 5.12.16 chart.grid.{x}.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist und backgroundColor einen anderen Wert als transparent hat.

### 5.12.17 chart.grid.{x}.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 5.12.18 chart.grid.{x}.shadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 5.12.19 chart.grid.{x}.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 5.12.20 chart.grid.{x}.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Koordinatensystems



### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

#### 5.12.21 chart.grid.{x}.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

#### 5.12.22 chart.grid.{x}.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.

ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentypen in grid und polar unterstützt. Weiterhin lassen sich Achsen über `axisPointer.axis` zuweisen.

- 'none'  
Durch nichts ausgelöst.

## Erlaubte Werte

- item
- axis
- none

### 5.12.23 chart.grid.{x}.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

`tooltip.axisPointer` ist wie syntaktischer Zucker von `axisPointer`-Einstellungen an Achsen (zum Beispiel `xAxis.axisPointer` oder `angleAxis.axisPointer`). Detailliertere Einstellungen können an `someAxis.axisPointer` vorgenommen werden. Im Normalfall ist die Verwendung von `tooltip.axisPointer` jedoch praktischer.

**Hinweis:** Konfigurationen an `tooltip.axisPointer` haben eine niedrigere Priorität als `someAxis.axisPointer`

### 5.12.24 chart.grid.{x}.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

### Erlaubte Werte

- line
- shadow
- none
- cross

### 5.12.25 chart.grid.{x}.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

### Erlaubte Werte

- x
- y
- radius
- angle
- auto

### 5.12.26 chart.grid.{x}.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.12.27 chart.grid.{x}.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.12.28 chart.grid.{x}.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des axisPointers.

### 5.12.29 chart.grid.{x}.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.12.30 chart.grid.{x}.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.12.31 chart.grid.{x}.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

**Rückgabewert:**

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.12.32 chart.grid.{x}.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 5.12.33 chart.grid.{x}.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.12.34 chart.grid.{x}.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### 5.12.35 chart.grid.{x}.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.12.36 chart.grid.{x}.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.12.37 chart.grid.{x}.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.12.38 chart.grid.{x}.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.12.39 chart.grid.{x}.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.12.40 chart.grid.{x}.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.12.41 chart.grid.{x}.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

#### 5.12.42 chart.grid.{x}.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 5.12.43 chart.grid.{x}.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.12.44 chart.grid.{x}.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.12.45 chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.12.46 chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.12.47 chart.grid.{x}.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.12.48 chart.grid.{x}.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.12.49 chart.grid.{x}.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.12.50 chart.grid.{x}.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.12.51 chart.grid.{x}.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.12.52 chart.grid.{x}.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.12.53 chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.12.54 chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.12.55 chart.grid.{x}.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' gesetzt ist.

### 5.12.56 chart.grid.{x}.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.12.57 chart.grid.{x}.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.12.58 chart.grid.{x}.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.12.59 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.12.60 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.12.61 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.12.62 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.12.63 chart.grid.{x}.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.12.64 chart.grid.{x}.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.12.65 chart.grid.{x}.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.12.66 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.12.67 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.12.68 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.12.69 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.12.70 chart.grid.{x}.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.12.71 chart.grid.{x}.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.



### 5.12.72 chart.grid.{x}.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.12.73 chart.grid.{x}.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 5.12.74 chart.grid.{x}.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.12.75 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.12.76 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.12.77 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.12.78 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.12.79 chart.grid.{x}.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.12.80 chart.grid.{x}.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

### 5.12.81 chart.grid.{x}.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

#### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

#### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

#### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.



Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.12.82 chart.grid.{x}.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.12.83 chart.grid.{x}.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.12.84 chart.grid.{x}.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.12.85 chart.grid.{x}.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.12.86 chart.grid.{x}.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 5.12.87 chart.grid.{x}.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.12.88 chart.grid.{x}.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.12.89 chart.grid.{x}.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.12.90 chart.grid.{x}.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.12.91 chart.grid.{x}.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.12.92 chart.grid.{x}.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.12.93 chart.grid.{x}.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.12.94 chart.grid.{x}.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.12.95 chart.grid.{x}.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.12.96 chart.grid.{x}.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.12.97 chart.grid.{x}.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.12.98 chart.grid.{x}.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.12.99 chart.grid.{x}.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.12.100 chart.grid.{x}.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.12.101 chart.grid.{x}.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

## 5.13 ECharts (4) Legend

### 5.13.1 chart.legend.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Legendenkomponente.

Die Legendenkomponente zeigt die Symbole, Farben und Namen verschiedener Serien an. Man kann Legenden anklicken um das Anzeigen von Serien im Diagramm umzuschalten.

In EChart 3 kann eine einzelne ECharts-Instanz mehrere Legendenkomponenten enthalten, was das Layout solcher vereinfacht.

Gibt es zu viele Legendeneinträge, so sind vertikal oder horizontal scrollbare Legenden eine Option, um sie zu paginieren. Bitte legend.type betrachten.

## 5.13.2 chart.legend.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Typ der Legende. Optionale Werte:

- 'plain': Einfache Legende (Standard)
- 'scroll': Scrollbare Legende. Hilfreich wenn zu viele Legendeneinträge angezeigt werden müssen.

Wenn 'scroll' verwendet wird, können die untenstehenden Optionen für detailliertere Konfiguration verwendet werden:

- legend.scrollDataIndex
- legend.pageButtonItemGap
- legend.pageButtonGap
- legend.pageButtonPosition
- legend.pageFormatter
- legend.pagerIcons
- legend.pagerIconColor
- legend.pagerIconInactiveColor
- legend.pagerIconSize
- legend.pageTextStyle
- legend.animation
- legend.animationDurationUpdate

### Erlaubte Werte

- plain
- scroll

### 5.13.3 chart.legend.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.13.4 chart.legend.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

### 5.13.5 chart.legend.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.13.6 chart.legend.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.13.7 chart.legend.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 5.13.8 chart.legend.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.13.9 chart.legend.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

## 5.13.10 chart.legend.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

### Erlaubte Werte

- {percent}
- {integer}

## 5.13.11 chart.legend.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Legenden-Komponente. Standardmäßig adaptiv.

### 5.13.12 chart.legend.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Legenden-Komponente. Standardmäßig adaptiv.

### 5.13.13 chart.legend.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Layout-Orientierung der Legende.

Erlaubte Werte

- horizontal
- vertical

### 5.13.14 chart.legend.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Ausrichtung von Legenden-Marker und -Text. Wird standardmäßig automatisch aus Position und Orientierung der Komponente berechnet. Wenn der Wert von left dieser Komponente 'right' und das Layout vertikal ist (orient ist 'vertical'), so würde sie auf 'right' ausgerichtet werden.

#### Erlaubte Werte

- auto
- left
- right

### 5.13.15 chart.legend.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der Legende um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:



```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.13.16 chart.legend.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Distanz zwischen den Legenden, horizontale Distanz im horizontalen Layout und vertikale Distanz im vertikalen layout.

### 5.13.17 chart.legend.itemWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 25 ]

Bildbreite des Legendensymbols.

### 5.13.18 chart.legend.itemHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Bildhöhe des Legendensymbols.

### 5.13.19 chart.legend.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob der Aspekt für Icons (von series.symbol oder nutzerdefiniertem legend.data.icon) in der Form von path:// behalten werden soll.

### 5.13.20 chart.legend.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatter wird verwendet, um das Label der Legende zu formatieren, welches String-Vorlagen und Callback-Funktionen unterstützt.

Beispiel:

```
// String-Vorlage benutzend, die Variable der Vorlage ist Legendenname {name}
formatter: 'Legend {name}'
// Callback-Funktion benutzend
formatter: function (name) {
  return 'Legend ' + name;
}
```

### 5.13.21 chart.legend.selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ausgewählter Legendenmodus, welcher kontrolliert, ob das Anzeigen von Serien durch Klicken ihrer Legenden umgeschaltet werden kann. Dies ist standardmäßig möglich und durch setzen dieses Wertes auf false wird es verhindert.

Außerdem kann dieser Wert auf 'single' oder 'multiple' gesetzt werden, für Einzelauswahl oder Mehrfachauswahl.

### 5.13.22 chart.legend.inactiveColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Farbe der Legende, wenn diese nicht ausgewählt ist.

### 5.13.23 chart.legend.selected

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Statustabelle der ausgewählten Legenden.

Beispiel:

```
selected: {
  // ausgewählte 'series 1'
  'series 1': true,
  // nicht ausgewählte 'series 2'
  'series 2': false
}
```

### 5.13.24 chart.legend.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Legendentexts.

### 5.13.25 chart.legend.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.13.26 chart.legend.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 5.13.27 chart.legend.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 5.13.28 chart.legend.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.13.29 chart.legend.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.13.30 chart.legend.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

Erlaubte Werte

- left
- center
- right

### 5.13.31 chart.legend.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

**Erlaubte Werte**

- top
- middle
- bottom

**5.13.32 chart.legend.textStyle.lineHeight**

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```



### 5.13.33 chart.legend.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.13.34 chart.legend.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.13.35 chart.legend.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.13.36 chart.legend.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[Default-Wert: 0]

Rahmenradius des Textfragments.

### 5.13.37 chart.legend.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.13.38 chart.legend.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.13.39 chart.legend.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.13.40 chart.legend.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

#### 5.13.41 chart.legend.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

#### 5.13.42 chart.legend.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 5.13.43 chart.legend.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.13.44 chart.legend.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.13.45 chart.legend.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.13.46 chart.legend.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.13.47 chart.legend.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.13.48 chart.legend.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.13.49 chart.legend.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.13.50 chart.legend.textStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.13.51 chart.legend.tooltip

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Konfiguration für den Tooltip der Legende.

## 5.13.52 chart.legend.data.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Datenarray der Legende. Ein Element des Arrays ist normalerweise ein String, der den Namen einer Serie (oder, in einem Kuchendiagramm, den Namen eines einzelnen Datenwerts) repräsentiert. Die Legendenkomponente berechnet dann die zugehörige Farbe und das Icon automatisch anhand der Serie. Die Spezialstrings "" (null-String) oder '\n' (Zeilenumbruch-String) können benutzt werden, um eine neue Zeile zu erhalten.

Wird data nicht spezifiziert, so werden die Namen automatisch aus den Serien geholt. Bei den meisten Serien werden diese aus series.name oder dem Dimensionsnamen, der durch seriesName von series.encode spezifiziert wird, bezogen. Bei manchen Serientypen, zum Beispiel bei zu Kuchen- oder Trichterdiagrammen gehörigen, kommen sie aus dem Namensfeld von series.data.

Soll der Stil eines Elementes festgelegt werden, kann diese Konfiguration mit angegeben werden. In diesem Fall wird ein name-Attribut verwendet um den Namen der Serie anzugeben.

Beispiel:

```
data: [{
  name: 'series 1',
  // erzwinge als Icon einen Kreis
  icon: 'circle',
  // setze die Textfarbe auf Rot
  textStyle: {
    color: 'red'
  }
}]
```

## 5.13.53 chart.legend.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe der Legende, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 5.13.54 chart.legend.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe der Legende. Unterstützt das gleiche Farbformat wie backgroundColor.

### 5.13.55 chart.legend.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite der Legende.

### 5.13.56 chart.legend.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 5.13.57 chart.legend.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 5.13.58 chart.legend.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.13.59 chart.legend.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.13.60 chart.legend.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

### 5.13.61 chart.legend.scrollDataIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

dataIndex des linksobert angezeigten Eintrags.

Obwohl das Scrollen von Legendeneinträgen kontrolliert werden kann indem setOption aufgerufen wird und diese Eigenschaft gesetzt wird, wird empfohlen, dies nicht zu tun, falls es nicht nötig ist (setOption kann Zeitaufwändig sein) und stattdessen die Aktion legendScroll zu verwenden.

### 5.13.62 chart.legend.pageButtonItemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Abstand zwischen den Seitenknöpfen und dem Seiteninfotext.

### 5.13.63 chart.legend.pageButtonGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Abstand zwischen den Seitenknöpfen und den Legendeneinträgen.

### 5.13.64 chart.legend.pageButtonPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'end' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Position der Seitenknöpfe und der Seiteninfo.

Mögliche Werte:

- 'start': Links oder oben.
- 'end': Rechts oder unten

### 5.13.65 chart.legend.pageFormatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '{current}/{total}' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Formatierer der Seiteninfo. Steht standardmäßig auf '{current}/{total}', wobei {current} die aktuelle Seitennummer (mit 1 anfangend), und {total} die Gesamtzahl an Seiten ist.

Ist pageFormatter eine Funktion, so sollte sie einen String zurückgeben. Die Parameter sind:

```
{
  current: number
  total: number
}
```

### 5.13.66 chart.legend.pageIcons

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Icons der Seitenknöpfe.

### 5.13.67 chart.legend.pageIcons.horizontal

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Icons der Seitenknöpfe wenn legend.orient auf 'horizontal' steht.

Der Wert sollte ein Array, [Knopf für vorherige Seite, Knopf für nächste Seite] sein, standardmäßig ['M0,0L12,-10L12,10z', 'M0,0L-12,-10L-12,10z'].

Für jedes Element des Arrays:

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.13.68 chart.legend.pageIcons.vertical

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Icons der Seitenknöpfe wenn legend.orient auf 'vertical' steht.

Der Wert sollte ein Array, [Knopf für vorherige Seite, Knopf für nächste Seite] sein, standardmäßig ['M0,0L20,0L10,-20z', 'M0,0L20,0L10,20z'].

Für jedes Element des Arrays:

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.13.69 chart.legend.pageIconColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: '#2f4554' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Farbe der Seitenknöpfe.

### 5.13.70 chart.legend.pageIconInactiveColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#aaa' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Farbe der Seitenknöpfe, wenn sie inaktiv sind.

### 5.13.71 chart.legend.pageIconSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Größe der Seitenknöpfe. Kann eine Zahl sein oder ein Array wie [10, 3], welches [Breite, Höhe] repräsentiert.

### 5.13.72 chart.legend.pageTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Textstil der Seiteninfo.

### 5.13.73 chart.legend.pageTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

Textfarbe.

### 5.13.74 chart.legend.pageTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'

- 'oblique'

### Erlaubte Werte

- normal
- italic
- oblique

### 5.13.75 chart.legend.pageTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 5.13.76 chart.legend.pageTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.13.77 chart.legend.pageTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.13.78 chart.legend.pageTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.13.79 chart.legend.pageTextStyle.width

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 5.13.80 chart.legend.pageTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.13.81 chart.legend.pageTextStyle.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.13.82 chart.legend.pageTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.13.83 chart.legend.pageTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.13.84 chart.legend.pageTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.13.85 chart.legend.pageTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.13.86 chart.legend.pageTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.13.87 chart.legend.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob das Scrollen animiert sein soll.

### 5.13.88 chart.legend.animationDurationUpdate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 800 ]

Dauer der Animation des Scrollens.

## 5.14 ECharts (4) Parallel

### 5.14.1 chart.parallel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

### Einführung über parallele Koordinaten

Parallele Koordinaten sind ein häufig genutzter Weg um hochdimensionale Geometrie zu visualisieren und multivariate Daten zu analysieren.

Zum Beispiel, sei `series-parallel.data` die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
    lineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

Parallele Koordinaten werden oft verwendet um Mehrdimensionale Daten wie oben zusammenzufassen. Jede Achse steht für eine Dimension (also eine Spalte) und jede Linie steht für ein Datenelement. Daten können auf den Achsen mit dem brush ausgewählt werden.

### Kurzfassung der Konfiguration

Die grundlegende Konfiguration von parallelen Koordinaten läuft wie folgt ab:

```

option = {
  parallelAxis: [                                     // Definition der Achsen.
    {dim: 0, name: schema[0].text}, // Jede Achse hat ein 'dim'-Attribut, welches
    // für den Index der Dimension in den Daten steht.
    {dim: 1, name: schema[1].text},
    {dim: 2, name: schema[2].text},
    {dim: 3, name: schema[3].text},
    {dim: 4, name: schema[4].text},
    {dim: 5, name: schema[5].text},
    {dim: 6, name: schema[6].text},
    {dim: 7, name: schema[7].text,
      type: 'category', // Unterstützt auch Kategoriedaten.
      data: ['Exzellent', 'gut', 'leichte Verschmutzung', 'moderate
Verschmutzung', 'schwere Verschmutzung', 'massive Verschmutzung']
    }
  ],
  parallel: {                                         // Definition eines parallelen
  Koordinatensystems.
    left: '5%',                                       // Platzierung des parallelen
  Koordinatensystems.
    right: '13%',
    bottom: '10%',
    top: '20%',
    parallelAxisDefault: { // Ein Muster für die Achsendefinition, damit
  dieses nicht in 'parallelAxis' wiederholt werden muss.
      type: 'value',
      nameLocation: 'end',
      nameGap: 20
    }
  },
  series: [ // Die drei Serien, die sich das parallele
  Kordinatensystem teilen.
    {
      name: 'Beijing',
      type: 'parallel', // Der Typ dieser Serie ist 'parallel'.
      data: [
        [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
        [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Shanghai',
      type: 'parallel',
      data: [
        [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
        [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Guangzhou',

```

```

        type: 'parallel',
        data: [
            [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
            [5, 42, 24, 44, 0.76, 40, 16, 'exzellent'],
            ...
        ]
    }
]
};

```

Drei Komponenten sind hier involviert: `parallel`, `parallelAxis`, `series-parallel`

- `parallel`  
Diese Komponente ist das Koordinatensystem. Eine oder mehrere Serien (wie "Beijing", "Shanghai" und "Guangzhou" im obigen Beispiel) können sich ein Koordinatensystem teilen. Wie andere Koordinatensysteme können mehrere parallele Koordinatensysteme in einer ECharts-Instanz generiert werden. Das Setzen der Position wird auch hier durchgeführt.
- `parallelAxis`  
Dies ist die Achsenkonfiguration. Mehrere Achsen sind für parallele Koordinatensysteme notwendig. `parallelAxis.parallelIndex` wird verwendet um zu spezifizieren, zu welchem Koordinatensystem die Achse gehört. Standardmäßig wird das erste Koordinatensystem verwendet.
- `series-parallel`  
Dies ist die Definition der parallelen Serien, welche im parallelen Koordinatensystem gezeichnet werden.

### Anmerkungen und Best Practices

Wenn mehrere `parallelAxis` konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter `parallel.parallelAxisDefault` definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von `parallel.parallelAxisDefault` in `parallelAxis` eingefügt, um die finale Achsenkonfiguration zu erhalten.

### Falls die Daten zu groß sind und für schlechte Performanz sorgen

Es wird empfohlen, `series-parallel.lineStyle.width` auf 0.5 (oder weniger) zu setzen, was die Performanz signifikant erhöhen kann.

### Darstellung hochdimensionaler Daten

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann `parallel.axisExpandable` verwendet werden, um die Darstellung zu verbessern.

## 5.14.2 `chart.parallel.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 5.14.3 chart.parallel.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

## 5.14.4 chart.parallel.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.14.5 chart.parallel.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 5.14.6 chart.parallel.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.14.7 chart.parallel.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der rechten Seite des Containers.

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein.

### Erlaubte Werte

- {integer}
- {percent}

### 5.14.8 chart.parallel.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

### 5.14.9 chart.parallel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Komponente des parallelen Koordinatensystems. Standardmäßig adaptiv.

### 5.14.10 chart.parallel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Komponente des parallelen Koordinatensystems. Standardmäßig adaptiv.

### 5.14.11 chart.parallel.layout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Layout-Modi, deren optionale Werte sind:

- 'horizontal': Platziere jede Achse horizontal.
- 'vertical': Platziere jede Achse vertikal.

Erlaubte Werte

- horizontal
- vertical

### 5.14.12 chart.parallel.axisExpandable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann parallel.axisExpandable verwendet werden, um die Darstellung zu verbessern.

### 5.14.13 chart.parallel.axisExpandCenter

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Index der Achse, die zu Beginn als Zentrum der Expansion verwendet wird. Hat keinen standard-Wert und muss manuell zugewiesen werden.

Für weitere Informationen bitte an `parallel.axisExpandable` wenden.

#### 5.14.14 `chart.parallel.axisExpandCount`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Definiert, wie viele Achsen zu Beginn im expandierten Zustand sind. Es wird empfohlen, diesen Wert manuell den Dimensionen entsprechend zu setzen.

Für weitere Informationen bitte an `parallel.axisExpandCenter` und `parallel.axisExpandable` wenden.

#### 5.14.15 `chart.parallel.axisExpandWidth`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand in Pixeln zwischen zwei Achsen im expandierten Zustand.

Für weitere Informationen bitte an `parallel.axisExpandable` wenden.

### 5.14.16 chart.parallel.axisExpandTriggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'click' ]

Mögliche Werte:

- 'click': Löse Expansion durch Mausklicks aus.
- 'mousemove': Löse Expansion durch Mausbewegungen aus.

Erlaubte Werte

- click
- mousemove

### 5.14.17 chart.parallel.parallelAxisDefault.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere parallelAxis konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter parallel.parallelAxisDefault definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von parallel.parallelAxisDefault in parallelAxis eingefügt, um die finale Achsenkonfiguration zu erhalten.

### 5.14.18 chart.parallel.parallelAxisDefault.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

#### 5.14.19 chart.parallel.parallelAxisDefault.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

#### 5.14.20 chart.parallel.parallelAxisDefault.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

#### 5.14.21 chart.parallel.parallelAxisDefault.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

#### 5.14.22 chart.parallel.parallelAxisDefault.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens, verwendet standardmäßig axisLine.lineStyle.color

### 5.14.23 chart.parallel.parallelAxisDefault.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.14.24 chart.parallel.parallelAxisDefault.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

#### Erlaubte Werte

- normal
- bold

- bolder
- lighter
- {integer}

### 5.14.25 chart.parallel.parallelAxisDefault.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.14.26 chart.parallel.parallelAxisDefault.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 5.14.27 chart.parallel.parallelAxisDefault.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

#### 5.14.28 chart.parallel.parallelAxisDefault.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.14.29 chart.parallel.parallelAxisDefault.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.14.30 chart.parallel.parallelAxisDefault.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.14.31 chart.parallel.parallelAxisDefault.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.14.32 chart.parallel.parallelAxisDefault.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.14.33 chart.parallel.parallelAxisDefault.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.14.34 chart.parallel.parallelAxisDefault.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.14.35 chart.parallel.parallelAxisDefault.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.14.36 chart.parallel.parallelAxisDefault.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.14.37 chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.14.38 chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

## 5.14.39 chart.parallel.parallelAxisDefault.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.14.40 chart.parallel.parallelAxisDefault.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.14.41 chart.parallel.parallelAxisDefault.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

#### 5.14.42 chart.parallel.parallelAxisDefault.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 5.14.43 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 5.14.44 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 5.14.45 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

#### 5.14.46 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

#### 5.14.47 chart.parallel.parallelAxisDefault.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

#### 5.14.48 chart.parallel.parallelAxisDefault.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.



### 5.14.49 chart.parallel.parallelAxisDefault.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 5.14.50 chart.parallel.parallelAxisDefault.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 5.14.51 chart.parallel.parallelAxisDefault.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 5.14.52 chart.parallel.parallelAxisDefault.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.14.53 chart.parallel.parallelAxisDefault.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

#### 5.14.54 chart.parallel.parallelAxisDefault.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

#### 5.14.55 chart.parallel.parallelAxisDefault.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.14.56 chart.parallel.parallelAxisDefault.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.14.57 chart.parallel.parallelAxisDefault.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.14.58 chart.parallel.parallelAxisDefault.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.14.59 chart.parallel.parallelAxisDefault.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 5.14.60 chart.parallel.parallelAxisDefault.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

#### 5.14.61 chart.parallel.parallelAxisDefault.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

#### 5.14.62 chart.parallel.parallelAxisDefault.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.14.63 chart.parallel.parallelAxisDefault.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.14.64 chart.parallel.parallelAxisDefault.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.14.65 chart.parallel.parallelAxisDefault.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.14.66 chart.parallel.parallelAxisDefault.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.14.67 chart.parallel.parallelAxisDefault.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]



## Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.14.68 chart.parallel.parallelAxisDefault.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.14.69 chart.parallel.parallelAxisDefault.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.14.70 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 5.14.71 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 5.14.72 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.14.73 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 5.14.74 chart.parallel.parallelAxisDefault.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.14.75 chart.parallel.parallelAxisDefault.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.14.76 chart.parallel.parallelAxisDefault.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.14.77 chart.parallel.parallelAxisDefault.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.14.78 chart.parallel.parallelAxisDefault.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.14.79 chart.parallel.parallelAxisDefault.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.14.80 chart.parallel.parallelAxisDefault.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.14.81 chart.parallel.parallelAxisDefault.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Stil der Achsenlinie.

#### 5.14.82 chart.parallel.parallelAxisDefault.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Farbe der Achsenticks wird standardmäßig auf axisLine.lineStyle.color gesetzt.

#### 5.14.83 chart.parallel.parallelAxisDefault.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

#### 5.14.84 chart.parallel.parallelAxisDefault.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.14.85 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.14.86 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.14.87 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.14.88 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.14.89 chart.parallel.parallelAxisDefault.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.14.90 chart.parallel.parallelAxisDefault.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

#### 5.14.91 chart.parallel.parallelAxisDefault.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

#### 5.14.92 chart.parallel.parallelAxisDefault.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.14.93 chart.parallel.parallelAxisDefault.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.14.94 chart.parallel.parallelAxisDefault.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.14.95 chart.parallel.parallelAxisDefault.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.14.96 chart.parallel.parallelAxisDefault.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```

// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getFullYear());
  }
  return texts.join('/');
}

```

#### 5.14.97 chart.parallel.parallelAxisDefault.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

#### 5.14.98 chart.parallel.parallelAxisDefault.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.14.99 chart.parallel.parallelAxisDefault.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.14.100 chart.parallel.parallelAxisDefault.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

#### 5.14.101 chart.parallel.parallelAxisDefault.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

#### 5.14.102 chart.parallel.parallelAxisDefault.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.14.103 chart.parallel.parallelAxisDefault.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.14.104 chart.parallel.parallelAxisDefault.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:



```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.14.105 chart.parallel.parallelAxisDefault.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.14.106 chart.parallel.parallelAxisDefault.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.14.107 chart.parallel.parallelAxisDefault.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

#### 5.14.108 chart.parallel.parallelAxisDefault.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

#### 5.14.109 chart.parallel.parallelAxisDefault.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.14.110 chart.parallel.parallelAxisDefault.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.14.111 chart.parallel.parallelAxisDefault.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.14.112 chart.parallel.parallelAxisDefault.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

#### 5.14.113 chart.parallel.parallelAxisDefault.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

#### 5.14.114 chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

#### 5.14.115 chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

#### 5.14.116 chart.parallel.parallelAxisDefault.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.14.117 chart.parallel.parallelAxisDefault.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.14.118 chart.parallel.parallelAxisDefault.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

#### 5.14.119 chart.parallel.parallelAxisDefault.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 5.14.120 chart.parallel.parallelAxisDefault.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 5.14.121 chart.parallel.parallelAxisDefault.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 5.14.122 chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.



### 5.14.123 chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.14.124 chart.parallel.parallelAxisDefault.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

## 5.15 ECharts (4) Parallel Axis

### 5.15.1 chart.parallelAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Komponente ist die Koordinatenachse für ein paralleles Koordinatensystem.

## Einführung über parallele Koordinaten

Parallele Koordinaten sind ein häufig genutzter Weg um hochdimensionale Geometrie zu visualisieren und multivariate Daten zu analysieren.

Zum Beispiel, sei `series-parallel.data` die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
    lineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

Parallele Koordinaten werden oft verwendet um Mehrdimensionale Daten wie oben zusammenzufassen. Jede Achse steht für eine Dimension (also eine Spalte) und jede Linie steht für ein Datenelement. Daten können auf den Achsen mit dem brush ausgewählt werden.

## Kurzfassung der Konfiguration

Die grundlegende Konfiguration von parallelen Koordinaten läuft wie folgt ab:

```

option = {
  parallelAxis: [                                     // Definition der Achsen.
    {dim: 0, name: schema[0].text}, // Jede Achse hat ein 'dim'-Attribut, welches
    // für den Index der Dimension in den Daten steht.
    {dim: 1, name: schema[1].text},
    {dim: 2, name: schema[2].text},
    {dim: 3, name: schema[3].text},
    {dim: 4, name: schema[4].text},
    {dim: 5, name: schema[5].text},
    {dim: 6, name: schema[6].text},
    {dim: 7, name: schema[7].text,
      type: 'category', // Unterstützt auch Kategoriedaten.
      data: ['Exzellent', 'gut', 'leichte Verschmutzung', 'moderate
Verschmutzung', 'schwere Verschmutzung', 'massive Verschmutzung']
    }
  ],
  parallel: {                                         // Definition eines parallelen
  Koordinatensystems.
    left: '5%',                                       // Platzierung des parallelen
  Koordinatensystems.
    right: '13%',
    bottom: '10%',
    top: '20%',
    parallelAxisDefault: { // Ein Muster für die Achsendefinition, damit
  dieses nicht in 'parallelAxis' wiederholt werden muss.
      type: 'value',
      nameLocation: 'end',
      nameGap: 20
    }
  },
  series: [                                           // Die drei Serien, die sich das parallele
  Kordinatensystem teilen.
    {
      name: 'Beijing',
      type: 'parallel', // Der Typ dieser Serie ist 'parallel'.
      data: [
        [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
        [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Shanghai',
      type: 'parallel',
      data: [
        [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
        [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Guangzhou',

```

```

    type: 'parallel',
    data: [
      [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
      [5, 42, 24, 44, 0.76, 40, 16, 'exzellent'],
      ...
    ]
  }
]
};

```

Drei Komponenten sind hier involviert: `parallel`, `parallelAxis`, `series-parallel`

- `parallel`  
Diese Komponente ist das Koordinatensystem. Eine oder mehrere Serien (wie "Beijing", "Shanghai" und "Guangzhou" im obigen Beispiel) können sich ein Koordinatensystem teilen. Wie andere Koordinatensysteme können mehrere parallele Koordinatensysteme in einer ECharts-Instanz generiert werden. Das Setzen der Position wird auch hier durchgeführt.
- `parallelAxis`  
Dies ist die Achsenkonfiguration. Mehrere Achsen sind für parallele Koordinatensysteme notwendig. `parallelAxis.parallelIndex` wird verwendet um zu spezifizieren, zu welchem Koordinatensystem die Achse gehört. Standardmäßig wird das erste Koordinatensystem verwendet.
- `series-parallel`  
Dies ist die Definition der parallelen Serien, welche im parallelen Koordinatensystem gezeichnet werden.

### Anmerkungen und Best Practices

Wenn mehrere `parallelAxis` konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter `parallel.parallelAxisDefault` definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von `parallel.parallelAxisDefault` in `parallelAxis` eingefügt, um die finale Achsenkonfiguration zu erhalten.

### Falls die Daten zu groß sind und für schlechte Performanz sorgen

Es wird empfohlen, `series-parallel.lineStyle.width` auf 0.5 (oder weniger) zu setzen, was die Performanz signifikant erhöhen kann.

### Darstellung hochdimensionaler Daten

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann `parallel.axisExpandable` verwendet werden, um die Darstellung zu verbessern.

## 5.15.2 `chart.parallelAxis.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 5.15.3 chart.parallelAxis.dim

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dimensionsindex der Koordinatenachse.

Zum Beispiel, sei series-parallel.data die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
    lineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

dim definiert, welche Dimension (welche Reihe) von Daten auf diese Achse gemappt wird

Start bei 0. Zum Beispiel: Falls die dim der Koordinatenachse auf 1 steht, wird die zweite Datenreihe auf diese Achse gemappt.

## 5.15.4 chart.parallelAxis.parallelIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Wird verwendet um zu definieren, in welches Koordinatensystem die Achse gemappt wird.

Zum Beispiel:

```
myChart.setOption({
  parallel: [
    {...}, // Das erste parallele Koordinatensystem
    {...} // Das zweite parallele Koordinatensystem
  ],
  parallelAxis: [
    {parallelIndex: 1, ...}, // Die erste Koordinatenachse, auf das zweite
    parallele Koordinatensystem gemappt
    {parallelIndex: 0, ...}, // Die zweite Koordinatenachse, auf das erste
    parallele Koordinatensystem gemappt
    {parallelIndex: 1, ...}, // Die dritte Koordinatenachse, auf das zweite
    parallele Koordinatensystem gemappt
    {parallelIndex: 0, ...} // Die vierte Koordinatenachse, auf das erste
    parallele Koordinatensystem gemappt
  ],
  ...
});
```

Gibt es nur ein paralleles Koordinatensystem, muss dieser Wert nicht konfiguriert werden.

### 5.15.5 chart.parallelAxis.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht aktualisiert werden soll, während die brush-Auswahl auf einer Achse verwendet wird. Steht der Wert auf false, so wird die Ansicht aktualisiert nachdem die brush-Auswahl durchgeführt wurde.

Bei großen Datenmengen wird empfohlen, diesen Wert auf false zu setzen, um Effizienzprobleme zu vermeiden.

### 5.15.6 chart.parallelAxis.areaSelectStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsauswahl auf Achsen ist möglich. Hier sind einige Konfigurationsmöglichkeiten.

### 5.15.7 chart.parallelAxis.areaSelectStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Breite der Auswahlbox.

### 5.15.8 chart.parallelAxis.areaSelectStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite der Auswahlbox

### 5.15.9 chart.parallelAxis.areaSelectStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(160,197,232)' ]

Rahmenfarbe der Auswahlbox.

### 5.15.10 chart.parallelAxis.areaSelectStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(160,197,232)' ]

Füllfarbe der Auswahlbox

### 5.15.11 chart.parallelAxis.areaSelectStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Auswahlbox.

### 5.15.12 chart.parallelAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'category' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheiden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 5.15.13 chart.parallelAxis.name

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

### 5.15.14 chart.parallelAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 5.15.15 chart.parallelAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Textstil des Achsennamens.

### 5.15.16 chart.parallelAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 5.15.17 chart.parallelAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

Erlaubte Werte

- normal
- italic
- oblique

### 5.15.18 chart.parallelAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.15.19 chart.parallelAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.15.20 chart.parallelAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 5.15.21 chart.parallelAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.15.22 chart.parallelAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.15.23 chart.parallelAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.15.24 chart.parallelAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.15.25 chart.parallelAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.15.26 chart.parallelAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.15.27 chart.parallelAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.15.28 chart.parallelAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.15.29 chart.parallelAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.15.30 chart.parallelAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.15.31 chart.parallelAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.15.32 chart.parallelAxis.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.15.33 chart.parallelAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.15.34 chart.parallelAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.15.35 chart.parallelAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.15.36 chart.parallelAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.15.37 chart.parallelAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.15.38 chart.parallelAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.15.39 chart.parallelAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.15.40 chart.parallelAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.15.41 chart.parallelAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.15.42 chart.parallelAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 5.15.43 chart.parallelAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 5.15.44 chart.parallelAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 5.15.45 chart.parallelAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 5.15.46 chart.parallelAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.15.47 chart.parallelAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.15.48 chart.parallelAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 5.15.49 chart.parallelAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.15.50 chart.parallelAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.15.51 chart.parallelAxis.interval

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.15.52 `chart.parallelAxis.logBase`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.15.53 `chart.parallelAxis.silent`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 5.15.54 chart.parallelAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 5.15.55 chart.parallelAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Einstellungen, die zu axisLine gehören.

### 5.15.56 chart.parallelAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.15.57 chart.parallelAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.15.58 chart.parallelAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.15.59 chart.parallelAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.15.60 chart.parallelAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenlinie.

### 5.15.61 chart.parallelAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.15.62 chart.parallelAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.15.63 chart.parallelAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 5.15.64 chart.parallelAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.15.65 chart.parallelAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.15.66 chart.parallelAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.15.67 chart.parallelAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.15.68 chart.parallelAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.15.69 chart.parallelAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.15.70 chart.parallelAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.15.71 chart.parallelAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.15.72 chart.parallelAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.15.73 chart.parallelAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.15.74 chart.parallelAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.15.75 chart.parallelAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.15.76 chart.parallelAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, wird standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.15.77 chart.parallelAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.15.78 chart.parallelAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.15.79 chart.parallelAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.15.80 chart.parallelAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 5.15.81 chart.parallelAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.15.82 chart.parallelAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.15.83 chart.parallelAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.15.84 chart.parallelAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 5.15.85 chart.parallelAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.15.86 chart.parallelAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 5.15.87 chart.parallelAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.15.88 chart.parallelAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.15.89 chart.parallelAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.15.90 chart.parallelAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.15.91 chart.parallelAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.15.92 chart.parallelAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.15.93 chart.parallelAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.15.94 chart.parallelAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.15.95 chart.parallelAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.15.96 chart.parallelAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.15.97 chart.parallelAxis.axisLabel.fontSize

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.15.98 chart.parallelAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.15.99 chart.parallelAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

#### 5.15.100 chart.parallelAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.15.101 chart.parallelAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.15.102 chart.parallelAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.15.103 chart.parallelAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.15.104 chart.parallelAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.15.105 chart.parallelAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.15.106 chart.parallelAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.15.107 chart.parallelAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.15.108 chart.parallelAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.15.109 chart.parallelAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.15.110 chart.parallelAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.15.111 chart.parallelAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.15.112 chart.parallelAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.15.113 chart.parallelAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.15.114 chart.parallelAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.15.115 chart.parallelAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.15.116 chart.parallelAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.15.117 chart.parallelAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.15.118 chart.parallelAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

## 5.16 ECharts (4) Polar

### 5.16.1 chart.polar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Polarkoordinatensysteme können in Streu- und Liniendiagrammen verwendet werden. Jedes Polarkoordinatensystem hat eine angleAxis und eine radiusAxis.

### 5.16.2 chart.polar.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.16.3 chart.polar.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plazierte, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist

zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plaziert.

#### 5.16.4 chart.polar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

z hat eine niedrigere Priorität als zlevel und erstellt keine neuen Canvas-Elemente.

#### 5.16.5 chart.polar.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Position des Zentrums vom Polarkoordinatensystem, wobei der erste Wert die horizontale Position, und der zweite die vertikale Position ist.

Prozentwerte werden unterstützt. Wenn ein Prozentwert angegeben wird, ist er beim ersten Wert relativ zur Breite des Containers, und beim zweiten relativ zur Höhe des Containers.

#### **Beispiel:**

```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 5.16.6 chart.polar.radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Radius des Polarkoordinatensystems. Als Wert sind möglich:

- Integer: Spezifiziert den äußeren Radius direkt.
- String: Zum Beispiel, '20%' bedeutet, dass der äußere Radius 20% der Größe des Viewports ist (der kleine zwischen Breite und Höhe des Chart-Containers).
- Array<Integer|String>: Der erste Wert spezifiziert den inneren Radius und der zweite Wert spezifiziert den äußeren Radius. Jeder Wert folgt den oberen Definitionen.

### 5.16.7 chart.polar.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Polarkoordinatensystems

#### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

### 5.16.8 chart.polar.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

### 5.16.9 chart.polar.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentypern in grid und polar unterstützt. Weiterhin lassen sich Achsen über axisPointer.axis zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item

- axis
- none

### 5.16.10 chart.polar.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

tooltip.axisPointer ist wie syntaktischer Zucker von axisPointer-Einstellungen an Achsen (zum Beispiel xAxis.axisPointer oder angleAxis.axisPointer). Detailliertere Einstellungen können an someAxis.axisPointer vorgenommen werden. Im Normalfall ist die Verwendung von tooltip.axisPointer jedoch praktischer.

**Hinweis:** Konfigurationen an tooltip.axisPointer haben eine niedrigere Priorität als someAxis.axisPointer

### 5.16.11 chart.polar.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

## Erlaubte Werte

- line
- shadow
- none
- cross

### 5.16.12 chart.polar.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

## Erlaubte Werte

- x
- y
- radius
- angle
- auto

### 5.16.13 chart.polar.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

#### 5.16.14 chart.polar.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

#### 5.16.15 chart.polar.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des axisPointers.

#### 5.16.16 chart.polar.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.16.17 chart.polar.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.16.18 chart.polar.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.16.19 chart.polar.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 5.16.20 chart.polar.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.16.21 chart.polar.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### 5.16.22 chart.polar.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.16.23 chart.polar.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.16.24 chart.polar.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße

### 5.16.25 chart.polar.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.16.26 chart.polar.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.16.27 chart.polar.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.16.28 chart.polar.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.16.29 chart.polar.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.16.30 chart.polar.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.16.31 chart.polar.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.16.32 chart.polar.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.16.33 chart.polar.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.16.34 chart.polar.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.16.35 chart.polar.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.16.36 chart.polar.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.16.37 chart.polar.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.16.38 chart.polar.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.16.39 chart.polar.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.16.40 chart.polar.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.16.41 chart.polar.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.16.42 chart.polar.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'line' ist.

### 5.16.43 chart.polar.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.16.44 chart.polar.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.16.45 chart.polar.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.16.46 chart.polar.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 5.16.47 chart.polar.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 5.16.48 chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.16.49 chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.16.50 chart.polar.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.16.51 chart.polar.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.16.52 chart.polar.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.16.53 chart.polar.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 5.16.54 chart.polar.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 5.16.55 chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.16.56 chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.16.57 chart.polar.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.16.58 chart.polar.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

### 5.16.59 chart.polar.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.16.60 chart.polar.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.16.61 chart.polar.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.16.62 chart.polar.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.16.63 chart.polar.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.16.64 chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.16.65 chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.16.66 chart.polar.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.16.67 chart.polar.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

### 5.16.68 chart.polar.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

#### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

#### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

#### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.16.69 chart.polar.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.16.70 chart.polar.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.16.71 chart.polar.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.16.72 chart.polar.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.16.73 chart.polar.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 5.16.74 chart.polar.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.16.75 chart.polar.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

### 5.16.76 chart.polar.tooltip.textStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.16.77 chart.polar.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.16.78 chart.polar.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.16.79 chart.polar.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.16.80 chart.polar.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.16.81 chart.polar.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.16.82 chart.polar.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.16.83 chart.polar.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.16.84 chart.polar.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.16.85 chart.polar.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.16.86 chart.polar.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.16.87 chart.polar.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.16.88 chart.polar.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

## 5.17 ECharts (4) Radar

### 5.17.1 chart.radar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Koordinatensystem für Radardiagramme. Diese Komponente gleicht der Polar-Komponente in ECharts 2. Da die Polar-Komponente in ECharts 3 zur Komponente für standardmäßige Polarkoordinatensysteme umstrukturiert wurde, wurde diese Komponente zu radar umbenannt, um Verwirrung zu vermeiden.

Radarkoordinatensysteme unterscheiden sich von Polarkoordinatensystemen dadurch, dass jeder Achsenindikator des Radarkoordinatensystems eine eigene Dimension ist. Der Stil der Indikator-Koordinatenachse kann durch die folgenden Einstellungsoptionen konfiguriert werden, zusammen mit name, axisLine, axisTick, axisLabel, splitLine, splitArea.

## 5.17.2 chart.radar.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 5.17.3 chart.radar.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plaziert.

#### 5.17.4 chart.radar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

#### 5.17.5 chart.radar.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Position des Zentrums vom Polarkoordinatensystem, wobei der erste Wert die horizontale Position, und der zweite die vertikale Position ist.

Prozentwerte werden unterstützt. Wenn ein Prozentwert angegeben wird, ist er beim ersten Wert relativ zur Breite des Containers, und beim zweiten relativ zur Höhe des Containers.

#### **Beispiel:**

```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 5.17.6 chart.radar.radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '75%' ]

Radius des Radarkoordinatensystems. Als Wert sind möglich:

- Integer: Spezifiziert den äußeren Radius direkt.
- String: Zum Beispiel, '20%' bedeutet, dass der äußere Radius 20% der Größe des Viewports ist (der kleine zwischen Breite und Höhe des Chart-Containers).
- Array<Integer|String>: Der erste Wert spezifiziert den inneren Radius und der zweite Wert spezifiziert den äußeren Radius. Jeder Wert folgt den oberen Definitionen.

### 5.17.7 chart.radar.startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 90 ]

Der Startwinkel des Koordinatensystems, welcher der Winkel der ersten Indikatorachse ist.



### 5.17.8 chart.radar.name.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Radardiagramms.

### 5.17.9 chart.radar.name.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Name des Indikators angezeigt werden soll.

### 5.17.10 chart.radar.name.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Formatter des Indikatornamens, welcher String-Templates und Callback-Funktionen unterstützt. Siehe das folgende Beispiel

```
// Verwendung von String-Templates, die Template-Variable {value} sollte der Name des
Indikators sein
formatter: '{value}?'
// Verwendung von Callback-Funktionen, der erste Parameter ist der Name des
Indikators und der zweite Parameter ist das Konfigurationsobjekt des Indikators
formatter: function (value, indicator) {
    return '?' + value + '?';
}
```

### 5.17.11 chart.radar.name.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 5.17.12 chart.radar.name.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.17.13 chart.radar.name.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.17.14 chart.radar.name.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.17.15 chart.radar.name.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.17.16 chart.radar.name.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.17.17 chart.radar.name.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.17.18 chart.radar.name.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.17.19 chart.radar.name.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.17.20 chart.radar.name.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[Default-Wert: 0]

Rahmenradius des Textfragments.

### 5.17.21 chart.radar.name.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].

- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.17.22 chart.radar.name.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.17.23 chart.radar.name.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.17.24 chart.radar.name.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.17.25 chart.radar.name.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.17.26 chart.radar.name.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.



## Erlaubte Werte

- {percent}

## 5.17.27 chart.radar.name.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.17.28 chart.radar.name.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 5.17.29 chart.radar.name.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.17.30 chart.radar.name.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.17.31 chart.radar.name.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.17.32 chart.radar.name.textShadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.17.33 chart.radar.name.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.17.34 chart.radar.name.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.17.35 chart.radar.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Distanz zwischen Indikatorname und Achse.

### 5.17.36 chart.radar.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Segmente der Indikatorachse.

### 5.17.37 chart.radar.shape

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'polygon' ]

Rendertyp des Radars, welcher 'polygon' und 'circle' unterstützt.

Erlaubte Werte

- polygon
- circle

### 5.17.38 chart.radar.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Berechnung der Skalierung relativ zu Null verhindert werden soll. Ist der Wert auf true gesetzt, so müssen die Ticks im Koordinatensystem nicht zwingend die Null enthalten, was für gewöhnlich in Streudiagrammen mit zweifachen numerischen Wertachsen nützlich ist.

### 5.17.39 chart.radar.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 5.17.40 chart.radar.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```

{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}

```

### 5.17.41 chart.radar.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 5.17.42 chart.radar.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.17.43 chart.radar.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.17.44 chart.radar.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.17.45 chart.radar.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

#### 5.17.46 chart.radar.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

#### 5.17.47 chart.radar.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.17.48 chart.radar.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 5.17.49 chart.radar.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

### 5.17.50 chart.radar.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.17.51 chart.radar.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.17.52 chart.radar.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.17.53 chart.radar.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.17.54 chart.radar.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.17.55 chart.radar.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.17.56 chart.radar.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.17.57 chart.radar.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinie.

### 5.17.58 chart.radar.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc'] ]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```

splitLine: {
 LineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}

```

### 5.17.59 chart.radar.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 5.17.60 chart.radar.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.17.61 chart.radar.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.17.62 chart.radar.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.17.63 chart.radar.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.17.64 chart.radar.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.17.65 chart.radar.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.17.66 chart.radar.splitArea.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.17.67 chart.radar.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.17.68 chart.radar.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereich der Achse im Gitterbereich, wird standardmäßig nicht angezeigt.

### 5.17.69 chart.radar.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)','rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbararray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.17.70 chart.radar.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.17.71 chart.radar.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.17.72 chart.radar.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.17.73 chart.radar.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.17.74 chart.radar.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.17.75 chart.radar.indicator.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Indikator des Radardiagramms, welcher benutzt wird um mehrere Variablen (Dimensionen) im Radardiagramm zuzuweisen. Hier ein Beispiel:

```
indicator: [
  { name: 'Sales (sales) ', max: 6500},
  { name: 'Administration (Administration) ', max: 16000, color: 'red'}, // Setze
den Indikator auf Rot
  { name: 'Information Technology (Information Technology) ', max: 30000},
  { name: 'Customer Support (Customer Support) ', max: 38000},
  { name: 'Development (Development) ', max: 52000},
  { name: 'Marketing (Marketing) ', max: 25000}
]
```

### 5.17.76 chart.radar.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert eine Farbe für den Indikator.

## 5.18 ECharts (4) Radius Axis

### 5.18.1 chart.radiusAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Radialachse des Polarkoordinatensystems.

### 5.18.2 chart.radiusAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.18.3 chart.radiusAxis.polarIndex

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Radialachse im Polarkoordinatensystem. Ist standardmäßig die erste Achse.

#### 5.18.4 chart.radiusAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheiden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 5.18.5 chart.radiusAxis.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

### 5.18.6 chart.radiusAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 5.18.7 chart.radiusAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 5.18.8 chart.radiusAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens, verwendet standardmäßig axisLine.lineStyle.color

### 5.18.9 chart.radiusAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.18.10 chart.radiusAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.18.11 chart.radiusAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.18.12 chart.radiusAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 5.18.13 chart.radiusAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.18.14 chart.radiusAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.18.15 chart.radiusAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.18.16 chart.radiusAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.18.17 chart.radiusAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.18.18 chart.radiusAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.18.19 chart.radiusAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.18.20 chart.radiusAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.18.21 chart.radiusAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.18.22 chart.radiusAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.18.23 chart.radiusAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.18.24 chart.radiusAxis.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.18.25 chart.radiusAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.26 chart.radiusAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.27 chart.radiusAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.18.28 chart.radiusAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.18.29 chart.radiusAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.18.30 chart.radiusAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.18.31 chart.radiusAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.18.32 chart.radiusAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.18.33 chart.radiusAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.18.34 chart.radiusAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 5.18.35 chart.radiusAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 5.18.36 chart.radiusAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 5.18.37 chart.radiusAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 5.18.38 chart.radiusAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.18.39 chart.radiusAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.18.40 chart.radiusAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 5.18.41 chart.radiusAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.18.42 chart.radiusAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.18.43 chart.radiusAxis.interval

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

#### 5.18.44 `chart.radiusAxis.logBase`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

#### 5.18.45 `chart.radiusAxis.silent`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 5.18.46 chart.radiusAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 5.18.47 chart.radiusAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Einstellungen, die zu axisLine gehören.

#### 5.18.48 chart.radiusAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

#### 5.18.49 chart.radiusAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.18.50 chart.radiusAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.18.51 chart.radiusAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.18.52 chart.radiusAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenlinie.

### 5.18.53 chart.radiusAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.18.54 chart.radiusAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.18.55 chart.radiusAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 5.18.56 chart.radiusAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.57 chart.radiusAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.58 chart.radiusAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.59 chart.radiusAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.60 chart.radiusAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.18.61 chart.radiusAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.18.62 chart.radiusAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.18.63 chart.radiusAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.18.64 chart.radiusAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.18.65 chart.radiusAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.18.66 chart.radiusAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.18.67 chart.radiusAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 5.18.68 chart.radiusAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.18.69 chart.radiusAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.18.70 chart.radiusAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.18.71 chart.radiusAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.72 chart.radiusAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.73 chart.radiusAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.74 chart.radiusAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.75 chart.radiusAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.18.76 chart.radiusAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 5.18.77 chart.radiusAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.18.78 chart.radiusAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 5.18.79 chart.radiusAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.18.80 chart.radiusAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.18.81 chart.radiusAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.18.82 chart.radiusAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.18.83 chart.radiusAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

#### 5.18.84 chart.radiusAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

#### 5.18.85 chart.radiusAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.18.86 chart.radiusAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.18.87 chart.radiusAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.18.88 chart.radiusAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.18.89 chart.radiusAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.18.90 chart.radiusAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.18.91 chart.radiusAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

#### 5.18.92 chart.radiusAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.18.93 chart.radiusAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.18.94 chart.radiusAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.18.95 chart.radiusAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.18.96 chart.radiusAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.18.97 chart.radiusAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.18.98 chart.radiusAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.18.99 chart.radiusAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.18.100 chart.radiusAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.18.101 chart.radiusAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.18.102 chart.radiusAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.103 chart.radiusAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.104 chart.radiusAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.18.105 chart.radiusAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.18.106 chart.radiusAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.18.107 chart.radiusAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.18.108 chart.radiusAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.18.109 chart.radiusAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.18.110 chart.radiusAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.18.111 chart.radiusAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.18.112 chart.radiusAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.18.113 chart.radiusAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.18.114 chart.radiusAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.18.115 chart.radiusAxis.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 5.18.116 chart.radiusAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 5.18.117 chart.radiusAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.18.118 chart.radiusAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.119 chart.radiusAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.120 chart.radiusAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.121 chart.radiusAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.122 chart.radiusAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.18.123 chart.radiusAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.18.124 chart.radiusAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.18.125 chart.radiusAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.18.126 chart.radiusAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 5.18.127 chart.radiusAxis.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)','rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.18.128 chart.radiusAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.129 chart.radiusAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.130 chart.radiusAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.131 chart.radiusAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.132 chart.radiusAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.18.133 chart.radiusAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 5.18.134 chart.radiusAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.18.135 chart.radiusAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.18.136 chart.radiusAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.18.137 chart.radiusAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.18.138 chart.radiusAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.18.139 chart.radiusAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.18.140 chart.radiusAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.18.141 chart.radiusAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 5.18.142 chart.radiusAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 5.18.143 chart.radiusAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.18.144 chart.radiusAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 5.18.145 chart.radiusAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.18.146 chart.radiusAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.18.147 chart.radiusAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.18.148 chart.radiusAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.18.149 chart.radiusAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.150 chart.radiusAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.18.151 chart.radiusAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.18.152 chart.radiusAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.18.153 chart.radiusAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.



## 5.18.154 chart.radiusAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.18.155 chart.radiusAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 5.18.156 chart.radiusAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.18.157 chart.radiusAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.18.158 chart.radiusAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.18.159 chart.radiusAxis.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.18.160 chart.radiusAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.18.161 chart.radiusAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.162 chart.radiusAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.163 chart.radiusAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.164 chart.radiusAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.165 chart.radiusAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 5.18.166 chart.radiusAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

### 5.18.167 chart.radiusAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.18.168 chart.radiusAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.18.169 chart.radiusAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.170 chart.radiusAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.171 chart.radiusAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.172 chart.radiusAxis.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.173 chart.radiusAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.18.174 chart.radiusAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.18.175 chart.radiusAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.18.176 chart.radiusAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.177 chart.radiusAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.178 chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.18.179 chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 5.18.180 chart.radiusAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 5.18.181 chart.radiusAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.18.182 chart.radiusAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.18.183 chart.radiusAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 5.18.184 chart.radiusAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.18.185 chart.radiusAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 5.18.186 chart.radiusAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.18.187 chart.radiusAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.18.188 chart.radiusAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.18.189 chart.radiusAxis.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.18.190 chart.radiusAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.18.191 chart.radiusAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.18.192 chart.radiusAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.18.193 chart.radiusAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.18.194 chart.radiusAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.18.195 chart.radiusAxis.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.18.196 chart.radiusAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

## 5.19 ECharts (4) Series

### 5.19.1 chart.series.i[].activeOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Bei der brush-Auswahl werden asugewählte Linien auf diese Transparenz gesetzt (was diese Linien hervorheben kann).

### 5.19.2 chart.series.i[].animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 5.19.3 chart.series.i[].animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

#### 5.19.4 chart.series.i[].animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 5.19.5 chart.series.i[].animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {  
  // Dauer ist für spätere Daten länger  
  return idx * 100;  
}
```

### 5.19.6 chart.series.i[].animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {  
  // Dauer ist für spätere Daten länger  
  return idx * 100;  
}
```

### 5.19.7 chart.series.i[].animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.8 chart.series.i[].animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut



### 5.19.9 chart.series.i[].animationEasingUpdate.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Zeitverzögerung bevor die Animation startet. Callback-Funktionen können verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
    return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
    return (params.count - 1 - params.index) * 30;
}
```

### 5.19.10 chart.series.i[].animationEasingUpdate.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung vor der Aktualisierung der Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben.

Zum Beispiel:

```
animationDelayUpdate: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 5.19.11 chart.series.i[].animationEasingUpdate.object

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn

- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.12 chart.series.i[[]].animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 5.19.13 chart.series.i[[]].animationType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'expansion' ]

Initialer Animationstyp.

- 'expansion' Standardmäßige expandierende Animation.
- 'scale' Skalierende Animation. Kann zusammen mit animationEasing='elasticOut' verwendet werden um einen Popup-Effekt zu haben.

#### Erlaubte Werte

- expansion
- scale

### 5.19.14 chart.series.i[].areaStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.15 chart.series.i[.].areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Der Stil des Bereichs.

### 5.19.16 chart.series.i[[]].areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.17 chart.series.i[[]].areaStyle.origin

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Ursprungsposition des Bereichs.

Standardmäßig ist der Bereich zwischen der Achsenlinie und den Daten der Bereich der gefüllt wird. Diese Einstellung erlaubt es, von den Daten zum Maximal- oder Minimalwert der Achse zu füllen.

Mögliche Werte:

- 'auto': Um von der Achsenlinie bis zu den Daten zu füllen;
- 'start': Um vom Minimalwert der Achse bis zu den Daten zu füllen;
- 'end': Um vom Maximalwert der Achse bis zu den Daten zu füllen;

Erlaubte Werte

- auto
- start

- end

### 5.19.18 chart.series.i[.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.19 chart.series.i[.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.20 chart.series.i[.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.21 chart.series.i[].areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.22 chart.series.i[].aspectScale

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.75 ]

Wird verwendet, um das Seitenverhältnis von geo zu skalieren.

Das endültige Seitenverhältnis berechnet sich aus:  $\text{geoBoundingRect.width} / \text{geoBoundingRect.height} * \text{aspectScale}$ .



### 5.19.23 chart.series.i[].avoidLabelOverlap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Strategie zum Vermeiden sich überlappender Beschriftungen aktiviert werden soll. Standardmäßig aktiviert, wodurch Beschriftungen auseinanderbewegt werden, wenn sie sich überlappen.

### 5.19.24 chart.series.i[].axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.25 chart.series.i[].axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.26 chart.series.i[].axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.27 chart.series.i[].axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.28 chart.series.i[].axisLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe

### 5.19.29 chart.series.i[].axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.30 chart.series.i[].axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.31 chart.series.i[].axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.32 chart.series.i[].axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.19.33 chart.series.i[].axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer der Beschriftung der Skala, unterstützt String-Templates und Callback-Funktionen. Beispiel:

```
// Verwendung von String-Templates. Die Template-Variable {value} steht für den
// standardmäßigen Text der Beschriftung.
formatter: '{value} kg'

// Verwendung von Callback-Funktionen. Der Funktionsparameter ist der Wert der Skala.
formatter: function (value) {
    return value + 'km/h';
}
```

### 5.19.34 chart.series.i[].axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.35 chart.series.i[].axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.36 chart.series.i[].axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Beschriftung der Achsenticks.

### 5.19.37 chart.series.i[].axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.38 chart.series.i[].axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.39 chart.series.i[[]].axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.



### 5.19.40 chart.series.i[].axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.41 chart.series.i[].axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.42 chart.series.i[].axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.43 chart.series.i[].axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 5.19.44 chart.series.i[].axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.45 chart.series.i[].axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 5.19.46 chart.series.i[].axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 5.19.47 chart.series.i[].axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 5.19.48 chart.series.i[].axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

#### 5.19.49 chart.series.i[].axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

#### 5.19.50 chart.series.i[].axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.51 chart.series.i[].axisLine.lineStyle.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-wert: [[0.2, '#91c7ae'], [0.8, '#63869e'], [1, '#c23531']] ]

Die Achsenlinie des Messdiagramms kann in mehrere Segmente mit verschiedenen Farben aufgeteilt werden. Die Endposition und Farbe jedes Segments kann in einem Array dargestellt werden.

### 5.19.52 chart.series.i[].axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie des Messdiagramms.

### 5.19.53 chart.series.i[].axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.19.54 chart.series.i[].axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 5.19.55 chart.series.i[].axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.56 chart.series.i[].axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.57 chart.series.i[].axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.58 chart.series.i[].axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Linienbreite.

### 5.19.59 chart.series.i[].axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenlinie des Messdiagramms gehören.

### 5.19.60 chart.series.i[].axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie des Messdiagramms angezeigt werden soll.

### 5.19.61 chart.series.i[].axisTick.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 8 ]

Die Länge der Ticklinie, kann ein Prozentwert relativ zu radius sein.

### 5.19.62 chart.series.i[].axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.63 chart.series.i[[]].axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenticks.

### 5.19.64 chart.series.i[].axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.65 chart.series.i[].axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.66 chart.series.i[].axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.67 chart.series.i[].axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.68 chart.series.i[].axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.69 chart.series.i[].axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.70 chart.series.i[].axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.19.71 chart.series.i[].axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Ticklinientyp.

### 5.19.72 chart.series.i[].axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Skala angezeigt werden soll.

### 5.19.73 chart.series.i[].axisTick.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Anzahl der durch Achsenticks getrennten Segmente zwischen den Trennlinien.

### 5.19.74 chart.series.i[].barCategoryGap

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '20%' ]

Der Säulenabstand einer einzelnen Serie, standardmäßig 20% des Kategorienabstands, kann als fester Wert gesetzt werden.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie im Koordinatensystem gesetzt werden, und wird dann von allen 'bar'-Serien im Koordinatensystem übernommen.

### 5.19.75 chart.series.i[].barGap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30% ]

Der Abstand zwischen Säulen verschiedener Serien, ist ein Prozentwert wie '30%', was für 30% der Breite der Säule steht.

Setzen von barGap auf '-100%' lässt Säulen verschiedener Serien überlappen, was nützlich ist, wenn eine Säulenserie als Hintergrund dienen soll.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie des Koordinatensystems gesetzt werden, und wird dann von allen 'bar'-Serien im koordinatensystem übernommen.

### 5.19.76 chart.series.i[].barMaxWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Maximale Breite der Säulen. Absolute Werte (wie 10) und Prozentwerte (wie '20%', nach der Breite des Diagramms) können verwendet werden. Adaptiv wenn nicht gesetzt.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie im Koordinatensystem gesetzt werden, und wird dann von allen 'bar'-Serien im Koordinatensystem übernommen.

### 5.19.77 chart.series.i[].barMinHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Minimalbreite der Säule. Kann verwendet werden, um die Situation zu vermeiden, dass Interaktionen durch Säulen mit zu kleinen Datenwerten beeinträchtigt werden.

### 5.19.78 chart.series.i[].barMinWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert minimale Breite der Säulen. Absolute Werte (wie 10) oder Prozentwerte (wie '20%', nach der Breite des Diagramms) können verwendet werden. Passt sich standardmäßig automatisch an.



## Erlaubte Werte

- {integer}
- {percent}

### 5.19.79 chart.series.i[].barWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Breite der Säule. Adaptiv wenn nicht spezifiziert.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie des Koordinatensystems gesetzt werden, und wird dann von allen 'bar'-Serien im koordinatensystem übernommen.

### 5.19.80 chart.series.i[].blurSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Verwischungsgröße der Datenpunkte. Ist gültig wenn coordinateSystem den Wert 'geo' hat.

### 5.19.81 chart.series.i[].bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der Unterseite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein, oder ein prozentualer Wert relativ zur Containerbreite wie '20%'.

Erlaubte Werte

- {integer}
- {percent}

### 5.19.82 chart.series.i[[]].boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['10%', '10%'] ]

Der Grenzabstand der Richtung, die Orthogonal zur KoordinatenAchse im Diagramm ist. Wird gesetzt, um die Position des Diagramms anzupassen, und es im Mittelpunkt des Bildschirms statt des oberen oder unteren Bereichs zu halten.

Erlaubte Werte

- {integer}
- {percent}

### 5.19.83 chart.series.i[[]].boundingCoords

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zweidimensionales Array. Definiert die Koordinaten von oben-links und unten-rechts in der Layout-Box

```
// Eine komplette Weltkarte
map: 'world',
left: 0, top: 0, right: 0, bottom: 0,
boundingCoords: [
  // [Längengrad, Breitengrad] der Ecke oben-links
  [-180, 90],
  // [Längengrad, Breitengrad] der Ecke unten-rechts
  [180, -90]
],
```

#### 5.19.84 chart.series.i[].boxWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [7, 50] ]

Obere und Untere Grenze der Größe der Box. Das Array hat die Form [min, max].

Möglich sind absolute Pixelwerte, wie [7, 50], oder Prozentwerte, wie ['40%', '90%']. Die Prozentwerte sind hierbei relativ zur maximal möglichen Breite.

#### 5.19.85 chart.series.i[].breadcrumb.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der Unterseite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerhöhe wie '20%'.

Erlaubte Werte

- {integer}
- {percent}

### 5.19.86 chart.series.i[].breadcrumb.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(255,255,255,0.7) ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.87 chart.series.i[].breadcrumb.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

## Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.88 chart.series.i[].breadcrumb.emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.89 chart.series.i[].breadcrumb.emphasis.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(0,0,0,0.7) ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.90 chart.series.i[].breadcrumb.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

#### 5.19.91 chart.series.i[].breadcrumb.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.19.92 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.93 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(150,150,150,1) ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.94 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.95 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.96 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.97 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.98 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.99 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.100 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.101 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Textfarbe.

### 5.19.102 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.103 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.104 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 5.19.105 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.106 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.107 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.108 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des hervorgehobenen Breadcrumb-Textes.

### 5.19.109 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.110 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.111 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.



## 5.19.112 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.113 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.114 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.115 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes.

### 5.19.116 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes.

## 5.19.117 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.19.118 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 5.19.119 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.120 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.121 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.verticalAlign`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im rich nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

### Erlaubte Werte

- top
- middle
- bottom

### 5.19.122 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### Erlaubte Werte

- {percent}

### 5.19.123 chart.series.i[].breadcrumb.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausgewähltes Element des Breadcrumbs.

### 5.19.124 chart.series.i[].breadcrumb.emptyItemWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 25 ]

Wenn der Breadcrumb keinen Inhalt hat, wird diese Minimalbreite verwendet.

### 5.19.125 chart.series.i[].breadcrumb.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 22 ]

Die Höhe des Breadcrumbs.

### 5.19.126 chart.series.i[.].breadcrumb.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(255,255,255,0.7) ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.127 chart.series.i[.].breadcrumb.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.128 chart.series.i[.].breadcrumb.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 5.19.129 chart.series.i[.].breadcrumb.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(0,0,0,0.7) ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.130 chart.series.i[.].breadcrumb.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

### 5.19.131 chart.series.i[.].breadcrumb.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.132 chart.series.i[.].breadcrumb.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.133 chart.series.i[.breadcrumb.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(150,150,150,1) ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.134 chart.series.i[.breadcrumb.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.135 chart.series.i[.breadcrumb.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.136 chart.series.i[.].breadcrumb.itemStyle.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.137 chart.series.i[.].breadcrumb.itemStyle.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
background-color: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.138 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.139 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.140 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.141 chart.series.i[.].breadcrumb.itemStyle.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Textfarbe.

### 5.19.142 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.143 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.144 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 5.19.145 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.146 chart.series.i[.].breadcrumb.itemStyle.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.147 chart.series.i[.].breadcrumb.itemStyle.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.148 chart.series.i[.].breadcrumb.itemStyle.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Breadcrumb-Textes.

### 5.19.149 chart.series.i[.breadcrumb.itemStyle.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.150 chart.series.i[.breadcrumb.itemStyle.textStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.151 chart.series.i[.].breadcrumb.itemStyle.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.152 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.153 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.154 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.155 chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes.

### 5.19.156 chart.series.i[].breadcrumb.itemStyle.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes.

### 5.19.157 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.158 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.159 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.160 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.161 chart.series.i[.].breadcrumb.itemStyle.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.162 chart.series.i[.].breadcrumb.itemStyle.textStyle.width

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 5.19.163 chart.series.i[.].breadcrumb.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Abstand zwischen der Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerbreite wie '20%'. Er kann auch 'left', 'center' oder 'right' sein.

Steht der Wert von left auf 'left', 'center' oder 'right', wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}



- {percent}
- left
- center
- right

### 5.19.164 chart.series.i[.].breadcrumb.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breadcrumb, zeigt den Pfad des momentanen Knotenpunkts an.

### 5.19.165 chart.series.i[.].breadcrumb.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Komponente und der rechten Seite des Containers.

Der Wert von right kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerbreite wie '20%'.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {integer}
- {percent}

### 5.19.166 chart.series.i[[]].breadcrumb.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Breadcrumb angezeigt werden soll.

### 5.19.167 chart.series.i[[]].breadcrumb.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerhöhe wie '20%'. Er kann auch 'top', 'middle' oder 'bottom' sein.

Steht der Wert von top auf 'top', 'middle' oder 'bottom', wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.19.168 chart.series.i[].calendarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Kalenderkoordinaten mit denen kombiniert werden soll, nützlich für mehrere Kalenderkoordinaten in einem Diagramm.

### 5.19.169 chart.series.i[].categories.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.170 chart.series.i[].categories.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.171 chart.series.i[.].categories.emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.172 chart.series.i[.].categories.emphasis.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.173 chart.series.i[.].categories.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

#### 5.19.174 chart.series.i[].categories.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.19.175 chart.series.i[].categories.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.176 chart.series.i[].categories.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.177 chart.series.i[].categories.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.178 chart.series.i[].categories.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.179 chart.series.i[].categories.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.180 chart.series.i[].categories.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.181 chart.series.i[].categories.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.182 chart.series.i[].categories.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.183 chart.series.i[].categories.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.184 chart.series.i[].categories.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.185 chart.series.i[].categories.emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.186 chart.series.i[].categories.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.187 chart.series.i[].categories.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.188 chart.series.i[].categories.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.189 chart.series.i[].categories.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.190 chart.series.i[].categories.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.191 chart.series.i[.].categories.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.192 chart.series.i[.].categories.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.193 chart.series.i[].categories.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Ausgewählten Knotens.

### 5.19.194 chart.series.i[].categories.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 5.19.195 chart.series.i[].categories.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.196 chart.series.i[].categories.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:



```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.197 chart.series.i[.categories.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.198 chart.series.i[.].categories.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.199 chart.series.i[].categories.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.200 chart.series.i[].categories.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.201 chart.series.i[].categories.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.202 chart.series.i[].categories.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.203 chart.series.i[].categories.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.204 chart.series.i[].categories.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.205 chart.series.i[].categories.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.206 chart.series.i[].categories.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.207 chart.series.i[].categories.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.208 chart.series.i[].categories.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.209 chart.series.i[].categories.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.210 chart.series.i[].categories.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.211 chart.series.i[].categories.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.212 chart.series.i[].categories.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausgewählte Kategorie.

### 5.19.213 chart.series.i[].categories.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.214 chart.series.i[].categories.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.215 chart.series.i[.categories.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.216 chart.series.i[.categories.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.217 chart.series.i[].categories.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Knoten dieser Kategorie.

### 5.19.218 chart.series.i[].categories.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.219 chart.series.i[].categories.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.220 chart.series.i[].categories.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.221 chart.series.i[].categories.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.222 chart.series.i[].categories.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.223 chart.series.i[.].categories.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.224 chart.series.i[.].categories.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.225 chart.series.i[].categories.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.226 chart.series.i[].categories.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.227 chart.series.i[].categories.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.228 chart.series.i[].categories.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.229 chart.series.i[].categories.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.230 chart.series.i[].categories.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.231 chart.series.i[].categories.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.232 chart.series.i[.categories.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.233 chart.series.i[.categories.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.234 chart.series.i[.].categories.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.235 chart.series.i[].categories.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.236 chart.series.i[].categories.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.237 chart.series.i[].categories.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Beschriftungen der Knoten dieser Kategorie.

### 5.19.238 chart.series.i[].categories.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 5.19.239 chart.series.i[].categories.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.240 chart.series.i[].categories.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.241 chart.series.i[.].categories.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.242 chart.series.i[.].categories.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.243 chart.series.i[].categories.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.244 chart.series.i[].categories.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.245 chart.series.i[].categories.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.246 chart.series.i[].categories.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.247 chart.series.i[].categories.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.248 chart.series.i[].categories.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.249 chart.series.i[].categories.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.250 chart.series.i[].categories.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.251 chart.series.i[].categories.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.252 chart.series.i[].categories.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.253 chart.series.i[].categories.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.254 chart.series.i[].categories.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.255 chart.series.i[].categories.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.256 chart.series.i[].categories.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Kategorie, welcher mit der Legende und den Beschriftungen korrespondieren kann.

### 5.19.257 chart.series.i[].categories.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Kategorisierung der Knoten, welche optional ist. Müssen Knoten klassifiziert werden, so lässt sich die Kategorie jedes Knotens über data[i].category festlegen. Der Stil der Kategorie wird auch auf de Knoten der Kategorie angewendet. Kategorien können auch in der Legende verwendet werden.

### 5.19.258 chart.series.i[].categories.symbol

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Knoten dieser Kategorie.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

#### 5.19.259 chart.series.i[].categories.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 5.19.260 chart.series.i[].categories.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 5.19.261 chart.series.i[].categories.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 5.19.262 chart.series.i[].categories.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße der Knoten dieser Kategorie. Kann auf eine einzelne Zahl wie 10 gesetzt werden, oder auf ein Array um Breite und Höhe darzustellen. Zum Beispiel, [20, 10] steht für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

### 5.19.263 chart.series.i[.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Zentrale Position des Diagramms, der erste Wert steht für die horizontale, der zweite für die vertikale Position.

Prozentwerte werden unterstützt. Werden Prozentwerte angegeben, so sind diese relativ zu jeweils der Containerbreite und Containerhöhe.

#### Beispiel:

```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 5.19.264 chart.series.i[.childrenVisibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: null ]

Kinder werden nicht angezeigt wenn die Größe des Bereichs eines Knotens kleiner als dieser Wert ist (Einheit: Quadratpixel).

Durch diese Einstellung können die Details von Knoten versteckt werden, wenn der rechteckige Bereich nicht groß genug ist. Wenn Nutzer auf den Knoten hereinzoomen, werden die Kindknoten angezeigt wenn der Bereich größer als der angegebene Wert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann das `childrenVisibleMin` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.265 `chart.series.i[].circular.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfiguration des zirkulären Layouts.

### 5.19.266 `chart.series.i[].circular.rotateLabel`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen automatisch gedreht werden sollen.

### 5.19.267 chart.series.i[].clipOverflow

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der überstehende Teil abgeschnitten werden soll, standardmäßig wird abgeschnitten.

### 5.19.268 chart.series.i[].clockwise

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob im Uhrzeigersinn gezeichnet werden soll.

### 5.19.269 chart.series.i[].colorAlpha

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Transparenzraten (Alpha-Kanal) der Serie an. Akzeptiert Werte von 0 bis 1.

Der Wert kann zum Beispiel [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorAlpha` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.270 `chart.series.i[].colorMappingBy`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'index' ]

Spezifiziert die Regel, nach der jeder Knoten seine Farbe aus der Farbliste erhält. Mögliche Werte:

- 'value':

Mappt `series-treemap.data.value` auf die Farben.

So zeigen die Farben der Knoten ihre Werte an.

`visualDimension` kann verwendet werden, um zu bestimmen, welche Datendimension für das visuelle Mapping verwendet wird.

- 'index':

Mappt den Index (Ordinalzahl) des Knotens auf die Farben. Dadurch bekommt in einer Ebene der erste Knoten die erste Farbe der Liste und der zweite Knoten die zweite Farbe.

So können aufeinanderfolgende Knoten durch ihre Farben unterschieden werden.

- 'id':

Mappt `series-treemap.data.id` auf die Farben.

Da die `id` verwendet wird um die Knoten zu identifizieren, behalten die Knoten so ihre ursprüngliche Farbe, wenn der Baum durch `setOption` verändert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

## Erlaubte Werte

- value
- index
- id

### 5.19.271 `chart.series.i[].colorSaturation`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Sättigungswerten der Serie an. Akzeptiert Werte von 0 bis 1.

Zum Beispiel kann `colorSaturation` [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.272 `chart.series.i[].connectNulls`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linie über Nullpunkte verbunden werden soll.

### 5.19.273 chart.series.i[].coordinateSystem

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cartesian2d' ]

Das Koordinatensystem, das für die Serie verwendet wird. Mögliche Werte sind:

- 'cartesian2d'  
Verwendet ein zweidimensionales, rechteckiges Koordinatensystem (auch kartesisches Koordinatensystem genannt), wobei die zugehörigen Achsenkomponenten über xAxisIndex und yAxisIndex zugewiesen werden.
- 'polar'  
Verwendet ein Polarkoordinatensystem, wobei die zugehörige Polarkoordinaten-Komponente über polarIndex zugewiesen wird.

#### Erlaubte Werte

- cartesian2d
- polar

### 5.19.274 chart.series.i[].cursor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'pointer' ]

Der Mausstil wenn die Maus auf ein Element zeigt, gleicht der cursor-Eigenschaft in CSS.

### 5.19.275 chart.series.i[].datasetIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Ist series.data nicht spezifiziert, und dataset existiert, so verwendet die Serie dataset. dataSetIndex spezifiziert, welches dataset verwendet wird.

### 5.19.276 chart.series.i[].detail.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

## 5.19.277 chart.series.i[.detail.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

## 5.19.278 chart.series.i[.detail.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.279 chart.series.i[.detail.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.280 chart.series.i[].detail.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Textfarbe.

### 5.19.281 chart.series.i[].detail.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.282 chart.series.i[].detail.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Schriftgröße.

### 5.19.283 chart.series.i[[]].detail.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.284 chart.series.i[[]].detail.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.19.285 chart.series.i[.].detail.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.286 chart.series.i[.].detail.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.287 chart.series.i[].detail.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Details des Messdiagramms, welche zum Anzeigen von Daten verwendet werden.

### 5.19.288 chart.series.i[].detail.offsetCenter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, '40%'] ]

Die Offset-Position relativ zum Mittelpunkt des Messdiagramms. Das erste Element des Arrays ist der horizontale Offset, das zweite Element ist der vertikale Offset. Kann in absoluten Werten oder Prozentwerten relativ zum Radius des Messdiagramms angegeben werden.

## Erlaubte Werte

- {integer}
- {percent}

### 5.19.289 chart.series.i[.].detail.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.290 chart.series.i[.].detail.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.291 chart.series.i[.].detail.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.292 chart.series.i[].detail.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.293 chart.series.i[].detail.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.294 chart.series.i[].detail.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.295 chart.series.i[].detail.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Details angezeigt werden sollen.

### 5.19.296 chart.series.i[].detail.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.297 chart.series.i[].detail.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.298 chart.series.i[.].detail.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.299 chart.series.i[.].detail.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.300 chart.series.i[.].detail.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.301 chart.series.i[.].detail.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.302 chart.series.i[.].detail.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.303 chart.series.i[].dimensions

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

dimensions kann verwendet werden um Informationen für die Dimensionen von series.data oder dataset.source zu definieren.

Hinweis: Wird dataset verwendet, so können wir Dimensionsnamen in der ersten Reihe/Spalte von dataset.source bereitstellen, und müssen dimensions nicht hier spezifizieren. Wird dimensions jedoch hier spezifiziert, so wird ECharts keine Dimensionsnamen mehr aus der ersten Reihe/Spalte von dataset.source holen.

Zum Beispiel:

```
option = {
  dataset: {
    source: [
      // 'date', 'open', 'close', 'highest', 'lowest'
      [12, 44, 55, 66, 2],
      [23, 6, 16, 23, 1],
      ...
    ]
  },
  series: {
    type: 'xxx',
    // Spezifiziere die Dimensionsnamen, welche im Tooltip angezeigt werden.
    dimensions: ['date', 'open', 'close', 'highest', 'lowest']
  }
}
```

```

series: {
  type: 'xxx',
  dimensions: [
    null, // Soll eine Dimension nicht definiert werden, ist null
    // Typ der Dimension spezifizieren.
    // 'ordinal' wird immer in Strings verwendet.
    // Ist der Typ nicht spezifiziert, so errät ECharts den
    // Typ anhand der Daten
    {name: 'good', type: 'number'},
    'bad' // Ist gleich zu {name: 'bad'}.
  ]
}

```

Jeder Eintrag von dimensions kann sein:

- String, zum Beispiel 'someName', was gleich ist zu {name: 'someName'}.
- Object, dessen Attribute sein können:
  - name: String.
  - type: String, unterstützt:
    - number
    - float, welches Float64Array ist
    - int, welches Int32Array ist
    - ordinal, diskreter Wert, der meistens einen String darstellt.
    - time, Zeitwert, siehe data für das Format von Zeitwerten.
  - displayName: string, im Allgemeinen im Tooltip zur Anzeige der Dimension verwendet. Falls nicht spezifiziert, wird standardmäßig name verwendet.

Wenn dimensions spezifiziert ist, wird tooltip vertikal angezeigt, wodurch Dimensionsnamen besser dargestellt werden können. Ansonsten wird tooltip horizontal angezeigt.

### 5.19.304 chart.series.i[.].downplay.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'white' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.305 chart.series.i[].downplay.itemStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.306 chart.series.i[].downplay.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 5.19.307 chart.series.i[].downplay.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.308 chart.series.i[].downplay.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

#### Stil der Ringdiagramm-Sektoren

Der Stil kann gesetzt werden in `series.itemStyle` für alle Sektoren dieser Serie, in `series.levels.itemStyle` für die Sektoren einer Ebene oder in `series.data.itemStyle` für einen einzelnen Sektor. Wenn `series.data.itemStyle` definiert ist, überdeckt es die Definitionen von `series.itemStyle` und `series.levels.itemStyle`.

**Priorität: `series.data.itemStyle` > `series.levels.itemStyle` > `series.itemStyle`.**

In ECharts wird *emphasis* für die Stile von Objekten verwendet, auf die die Maus zeigt. Für Ringdiagramme gibt es zwei zusätzliche States: *highlight* um Elemente hervorzuheben, die zu dem ausgewählten gehören, und *downplay* für nicht zugehörige.

### 5.19.309 chart.series.i[].downplay.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.310 chart.series.i[].downplay.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.311 chart.series.i[.].downplay.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.312 chart.series.i[.].downplay.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.313 chart.series.i[.downplay.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.314 chart.series.i[.downplay.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Ausrichtung des Textes. Man bemerke, dass 'left' für die innere Seite und 'right' für die äußere Seite steht.

#### Erlaubte Werte

- left
- center
- right

### 5.19.315 chart.series.i[.downplay.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.316 chart.series.i[].downplay.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.317 chart.series.i[].downplay.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.318 chart.series.i[].downplay.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.319 chart.series.i[].downplay.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.320 chart.series.i[].downplay.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.321 chart.series.i[].downplay.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.322 chart.series.i[].downplay.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.323 chart.series.i[.].downplay.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.324 chart.series.i[.].downplay.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.325 chart.series.i[].downplay.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.326 chart.series.i[].downplay.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.327 chart.series.i[].downplay.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.328 chart.series.i[].downplay.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 5.19.329 chart.series.i[].downplay.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

label setzt den Textstil für alle Sektoren.

**Priorität: series.data.label > series.levels.label > series.label**

Textbeschriftung der Sektoren, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 5.19.330 chart.series.i[.].downplay.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.331 chart.series.i[.].downplay.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.



### 5.19.332 chart.series.i[].downplay.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.333 chart.series.i[].downplay.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.19.334 chart.series.i[.].downplay.label.rotate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'radial' ]

Ist der Wert eine Zahl, steht er für eine Rotation, von -90 bis 90 Grad, wobei positive Werte gegen den Uhrzeigersinn rotieren.

Außerdem kann der String 'radial', für radielle Rotation, oder 'tangential', für tangentielle Rotation, verwendet werden.

Der Wert ist standardmäßig 'radial'. Ist keine Rotation erwünscht, kann er auf 0 gesetzt werden.

#### Erlaubte Werte

- {integer}
- radial
- tangential

### 5.19.335 chart.series.i[.].downplay.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.336 chart.series.i[.].downplay.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.337 chart.series.i[].downplay.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.338 chart.series.i[].downplay.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.339 chart.series.i[].downplay.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 5.19.340 chart.series.i[.].downplay.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.341 chart.series.i[.].downplay.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 5.19.342 chart.series.i[].downplay.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.19.343 chart.series.i[].downplay.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 5.19.344 chart.series.i[].downplay.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.345 chart.series.i[].downplay.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.346 chart.series.i[].downplay.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.19.347 chart.series.i[.].downplay.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.348 chart.series.i[.].downplay.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Elemente wenn die Maus auf ein Element zeigt, zu dem sie keine Beziehung haben. Siehe highlightPolicy



### 5.19.349 chart.series.i[].draggable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob Knoten gezogen werden können.

### 5.19.350 chart.series.i[].drillDownIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '?' ]

Markierung wenn in einen Knoten reingebohrt werden kann.

### 5.19.351 chart.series.i[].edgeLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.19.352 chart.series.i[.].edgeLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

## 5.19.353 chart.series.i[].edgeLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

## 5.19.354 chart.series.i[].edgeLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.355 chart.series.i[].edgeLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.356 chart.series.i[].edgeLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.19.357 chart.series.i[].edgeLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.358 chart.series.i[].edgeLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.359 chart.series.i[.edgeLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.360 chart.series.i[.edgeLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.361 chart.series.i[.edgeLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.362 chart.series.i[].edgeLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.363 chart.series.i[].edgeLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.364 chart.series.i[].edgeLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Kantenbeschriftungen.

### 5.19.365 chart.series.i[].edgeLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.366 chart.series.i[.].edgeLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

## 5.19.367 chart.series.i[.].edgeLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.368 chart.series.i[.edgeLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.369 chart.series.i[].edgeLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.370 chart.series.i[].edgeLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.371 chart.series.i[].edgeLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.372 chart.series.i[].edgeLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Kantenbeschriftung angezeigt werden soll.

### 5.19.373 chart.series.i[].edgeLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.374 chart.series.i[].edgeLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.375 chart.series.i[].edgeLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.376 chart.series.i[].edgeLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.377 chart.series.i[].edgeLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.378 chart.series.i[.edgeLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.379 chart.series.i[.edgeLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

## Erlaubte Werte

- top
- middle
- bottom

### 5.19.380 chart.series.i[.].edgeLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.381 chart.series.i[.].edges

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Alias von links.

### 5.19.382 chart.series.i[.].edgeSymbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['none', 'none'] ]

Symbole der beiden Enden der Kantenlinien.

Zum Beispiel:

```
edgeSymbol: ['circle', 'arrow']
```

#### Erlaubte Werte

- none
- circle
- arrow
- rect
- roundRect
- triangle
- diamond
- pin

### 5.19.383 chart.series.i[.].edgeSymbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe der Symbole an den Enden der Kantenlinien. Akzeptiert ein Array oder eine einzelne Zahl.

Zum Beispiel:

```
// Startsymbol hat Größe 5 und Endsymbol hat Größe 10
symbolSize: [5, 10],
// Beide haben Größe 10
symbolSize: 10
```

### 5.19.384 chart.series.i[].effect.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Farbe des Spezialeffekt-Symbols, welche standardmäßig identisch zu `lineStyle.color` ist.

### 5.19.385 chart.series.i[].effect.constantSpeed

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Falls die Bewegung des Spezialeffekt-Symbols eine konstante Geschwindigkeit hat, welche in Pixel pro Sekunde angegeben wird. Ist constantSpeed größer als 0, so wird period ignoriert.

### 5.19.386 chart.series.i[].effect.delay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verzögerung der Effektanimation. Kann Nummer oder Callback-Funktion sein.

### 5.19.387 chart.series.i[].effect.loop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Animation wiederholt werden soll.

### 5.19.388 chart.series.i[].effect.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Die Einstellungen für Spezialeffekte der Linien.

**Tips:** Alle Graphen mit Spureneffekten sollten auf eine individuelle Ebene gelegt werden, was bedeutet, dass ihr zlevel-Wert anders als der von andern sein muss. Die Animation dieser Ebene sollte währenddessen ausgeschaltet werden (animation: false). Ansonsten würden andere grafische Elemente in anderen Serien und die Animationsbeschriftung unnötige Geister produzieren.

## 5.19.389 chart.series.i[].effect.period

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 4 ]

Die Dauer des Spezialeffekts, in Sekunden angegeben.

## 5.19.390 chart.series.i[].effect.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Spezialeffekte angezeigt werden sollen.

## 5.19.391 chart.series.i[].effect.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'circle' ]

Das Symbol des Spezialeffekts.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

**Tip:** Der Winkel des Symbols ändert sich wenn die Tangente des Pfades sich ändert. Wird ein eigendefinierter Pfad verwendet, sollte sichergestellt werden, dass der Pfad nach oben ausgerichtet ist. So wird sichergestellt, dass das Symbol immer richtig ausgerichtet ist, wenn es sich auf dem Pfad bewegt.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

#### 5.19.392 chart.series.i[.].effect.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Die Größe des Spezialeffekt-Symbols, welche als einzelne Zahl, wie 10, gesetzt werden kann. Außerdem kann ein Array verwendet werden, um Breite und Höhe getrennt zu definieren. um Beispiel, [20, 10] steht für eine Breite von 20 und eine Höhe von 10.

### 5.19.393 chart.series.i[].effect.trailLength

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.2 ]

Die Länge der Spur des Spezialeffekts. Akzeptiert werden Werte von 0 bis 1. Die Spur wird länger je größer der Wert ist.

### 5.19.394 chart.series.i[].effectType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'ripple' ]

Effekttyp. Momentan wird nur der Kräuselungseffekt 'ripple' unterstützt.

Erlaubte Werte

- ripple

### 5.19.395 chart.series.i[].emphasis.edgeLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.396 chart.series.i[].emphasis.edgeLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.397 chart.series.i[].emphasis.edgeLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.398 chart.series.i[].emphasis.edgeLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.399 chart.series.i[].emphasis.edgeLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.400 chart.series.i[].emphasis.edgeLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.19.401 chart.series.i[].emphasis.edgeLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.402 chart.series.i[].emphasis.edgeLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.403 chart.series.i[].emphasis.edgeLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.404 chart.series.i[].emphasis.edgeLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.405 chart.series.i[.emphasis.edgeLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 5.19.406 chart.series.i[.emphasis.edgeLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.407 chart.series.i[].emphasis.edgeLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.408 chart.series.i[].emphasis.edgeLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der hervorgehobenen Kante.

### 5.19.409 chart.series.i[].emphasis.edgeLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.410 chart.series.i[].emphasis.edgeLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

### 5.19.411 chart.series.i[].emphasis.edgeLabel.rich.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

Label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

#### 5.19.412 chart.series.i[].emphasis.edgeLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.413 chart.series.i[].emphasis.edgeLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.414 chart.series.i[].emphasis.edgeLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.415 chart.series.i[].emphasis.edgeLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.416 chart.series.i[].emphasis.edgeLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Kantenbeschriftung angezeigt werden soll.

### 5.19.417 chart.series.i[].emphasis.edgeLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.418 chart.series.i[].emphasis.edgeLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.419 chart.series.i[].emphasis.edgeLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.420 chart.series.i[].emphasis.edgeLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.421 chart.series.i[].emphasis.edgeLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.422 chart.series.i[].emphasis.edgeLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.423 chart.series.i[].emphasis.edgeLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

#### 5.19.424 chart.series.i[].emphasis.edgeLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes

Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.425 chart.series.i[].emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.426 chart.series.i[].emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.427 chart.series.i[].emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.428 `chart.series.i[i].emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.429 chart.series.i[.].emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Knoten.

### 5.19.430 chart.series.i[].emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.431 chart.series.i[].emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 5.19.432 chart.series.i[].emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 5.19.433 chart.series.i[].emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.19.434 chart.series.i[].emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.435 chart.series.i[].emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.436 chart.series.i[].emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.437 chart.series.i[].emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.438 chart.series.i[].emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.439 chart.series.i[].emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.440 chart.series.i[].emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.441 chart.series.i[].emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.442 chart.series.i[].emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.443 chart.series.i[].emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 5.19.444 chart.series.i[].emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 5.19.445 chart.series.i[].emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.446 chart.series.i[].emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

#### 5.19.447 chart.series.i[].emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 5.19.448 chart.series.i[].emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

#### 5.19.449 chart.series.i[].emphasis.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

#### 5.19.450 chart.series.i[].emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des hervorgehobenen Knotens.

### 5.19.451 chart.series.i[].emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.452 chart.series.i[].emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.453 chart.series.i[].emphasis.label.position

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

#### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

#### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

#### 5.19.454 chart.series.i[].emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.19.455 chart.series.i[.].emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.456 chart.series.i[].emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.457 chart.series.i[].emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.458 chart.series.i[].emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.459 chart.series.i[].emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.460 chart.series.i[].emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.461 chart.series.i[].emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.462 chart.series.i[].emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.463 chart.series.i[].emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.464 chart.series.i[].emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.465 chart.series.i[].emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.466 chart.series.i[].emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.19.467 chart.series.i[].emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

## 5.19.468 chart.series.i[].emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.469 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.470 chart.series.i[.].emphasis.labelLine.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der visuellen Hilfslinie der Beschriftung. Steht label.position auf 'left' oder 'right', wird die Hilfslinie angezeigt.

### 5.19.471 chart.series.i[].emphasis.labelLine.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.472 chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.473 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowColor`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.474 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetX`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.475 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.476 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.477 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.478 chart.series.i[.emphasis.labelLine.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie im hervorgehobenen Status.

### 5.19.479 `chart.series.i[].emphasis.labelLine.emphasis.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die visuelle Hilfslinie im hervorgehobenen Zustand angezeigt werden soll.

### 5.19.480 `chart.series.i[].emphasis.labelLine.length`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des ersten Teils der hervorgehobenen visuellen Hilfslinie.

### 5.19.481 `chart.series.i[].emphasis.labelLine.lineStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.482 chart.series.i[.].emphasis.labelLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



**Schlagworte / Tags**

Linienstil der visuellen Hilfslinie.

### 5.19.483 chart.series.i[].emphasis.labelLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.484 chart.series.i[].emphasis.labelLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.485 chart.series.i[].emphasis.labelLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.486 chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.487 chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.488 chart.series.i[.emphasis.labelLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.489 chart.series.i[.emphasis.labelLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.490 chart.series.i[.emphasis.labelLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen visuellen Hilfslinie.

### 5.19.491 `chart.series.i[].emphasis.labelLine.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die hervorgehobene visuelle Hilfslinie angezeigt werden soll.

### 5.19.492 `chart.series.i[].emphasis.lineStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.493 chart.series.i[.].emphasis.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: 0.5 ]

Die Krümmung der Kanten im Graphen.

### 5.19.494 chart.series.i[].emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.19.495 chart.series.i[].emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.496 chart.series.i[].emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.497 chart.series.i[].emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.498 chart.series.i[].emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.499 chart.series.i[].emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 5.19.500 chart.series.i[].emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Elemente.

### 5.19.501 chart.series.i[].emphasis.upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 5.19.502 chart.series.i[].emphasis.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.503 chart.series.i[].emphasis.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.504 chart.series.i[].emphasis.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.505 chart.series.i[].emphasis.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.506 chart.series.i[].emphasis.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.507 chart.series.i[].emphasis.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 5.19.508 chart.series.i[].emphasis.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 5.19.509 chart.series.i[].emphasis.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.510 chart.series.i[].emphasis.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.511 chart.series.i[.emphasis.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.512 chart.series.i[.emphasis.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.513 chart.series.i[].emphasis.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.514 chart.series.i[].emphasis.upperLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.515 chart.series.i[].emphasis.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.516 chart.series.i[].emphasis.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen oberen Beschriftung.

### 5.19.517 chart.series.i[].emphasis.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 5.19.518 chart.series.i[].emphasis.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.519 chart.series.i[].emphasis.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.520 chart.series.i[.emphasis.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.521 chart.series.i[.emphasis.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.522 chart.series.i[].emphasis.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.523 chart.series.i[].emphasis.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.524 chart.series.i[].emphasis.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.525 chart.series.i[].emphasis.upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.526 chart.series.i[].emphasis.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.527 chart.series.i[].emphasis.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.528 chart.series.i[].emphasis.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.529 chart.series.i[].emphasis.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.530 chart.series.i[].emphasis.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.531 chart.series.i[].emphasis.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.532 chart.series.i[].emphasis.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.533 chart.series.i[].emphasis.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.534 chart.series.i[].emphasis.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.535 chart.series.i[.].encode

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert für jedes Element von data wohin es kodiert wird. Zum Beispiel:

```
option = {
  dataset: {
    source: [
      // Jede Säule ist eine Dimension.
      // Es gibt fünf Dimensionen: 0, 1, 2, 3, 4.
      [12, 44, 55, 66, 2],
      [23, 6, 16, 23, 1],
      ...
    ]
  },
  series: {
    type: 'xxx',
    encode: {
      x: [3, 1, 5],      // Dimension 3, 1, 5 ist auf X-Achse gemappt.
      y: 2,             // Dimension 2 ist auf Y-Achse gemappt.
      tooltip: [3, 2, 4] // Dimension 3, 2, 4 werden im Tooltip angezeigt.
    }
  }
}
```

Die Attribute von encode ändern sich je nach Typ des Koordinatensystems. Für cartesian2d können x und y definiert werden. Für polar können radius und angle definiert werden. Für geo können lng und lat definiert werden. Die Attribute tooltip und itemName (Name des Datenelements im tooltip) können immer definiert werden.

Wenn `dimensions` benutzt wird um Namen für bestimmte Dimensionen zu definieren, kann `encode` diese Namen direkt referenzieren. Zum Beispiel:

```
series: {
  type: 'xxx',
  dimensions: ['date', 'open', 'close', 'highest', 'lowest'],
  encode: {
    x: 'date',
    y: ['open', 'close', 'highest', 'lowest']
  }
}
```

Im besonderen kann in benutzerdefinierten Serien ein Attribut in `encode`, welches zu einer Achse gehört, auf null gesetzt werden, damit die Serie nicht von dieser Achse kontrolliert wird. Die Serientaten werden also nicht für die Länge der Achse betrachtet, und der `dataZoom` auf der Achse wird die Serie nicht filtern.

```
var option = {
  xAxis: {},
  yAxis: {},
  dataZoom: [{
    xAxisIndex: 0
  }, {
    yAxisIndex: 0
  }],
  series: {
    type: 'custom',
    renderItem: function (params, api) {
      return {
        type: 'circle',
        shape: {
          cx: 100, // x position ist immer 100
          cy: api.coord([0, api.value(0)])[1],
          r: 30
        },
        style: {
          fill: 'blue'
        }
      }
    };
  },
  encode: {
    // Dann wird die Serie nicht von der X-Achse
    // und dem zugehörigen dataZoom kontrolliert.
    x: -1,
    y: 1
  },
  data: [ ... ]
};
```

### 5.19.536 chart.series.i[].endAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: -45 ]

Der Endwinkel des Messdiagramms.

### 5.19.537 chart.series.i[].expandAndCollapse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Ein- und Ausklappen von Unterbäumen aktiviert ist, standardmäßig true. Da die Zeichenfläche limitiert ist, und ein Baum oft viele Knoten hat, können diese sich überlappen. Um dies zu vermeiden, können momentan unwichtige Unterbäume eingeklappt werden, bis sie wieder gebraucht werden.

**Hinweis: Wird ein nutzerdefiniertes Bild als Symbol eines Knotens definiert, ist es nicht möglich zwischen dem ein- und ausgeklappten Zustand durch die Füllfarbe zu unterscheiden. Das Hochladen von zwei Bildern, die die jeweiligen Zustände repräsentieren, wird momentan nicht unterstützt. Soll der Status des Knotens ersichtlich sein, wird also die Verwendung von regulären ECharts Tag-Typen, wie emptyCircle, empfohlen.**

### 5.19.538 chart.series.i[].focusNodeAdjacency

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ermöglicht dass, wenn die Maus auf eine Kante oder einen Knoten zeigt, auch mit diesem Element verbundene Kanten und Knoten hervorgehoben werden. Standardmäßig deaktiviert, kann manuell konfiguriert werden.

Mögliche Werte:

- false: Beim Zeigen auf eine Kante oder einen Knoten wird nur dieses Element hervorgehoben.
- true: Identisch zu 'allEdges'.
- 'allEdges': Beim Zeigen auf einen Knoten werden alle umgebenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.
- 'outEdges': Beim Zeigen auf einen Knoten werden alle ausgehenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.
- 'inEdges': Beim Zeigen auf einen Knoten werden alle eingehenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.

Erlaubte Werte

- {boolean}
- allEdges
- outEdges
- inEdges

### 5.19.539 chart.series.i[.].force.edgeLength

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Die Distanz zwischen zwei Knoten, die durch eine Kante verbunden sind. Diese Distanz wird auch durch repulsion beeinflusst.

Dieser Wert kann ein Array sein, um eine Spannweite an Kantenlängen darzustellen. In diesem Fall werden Kanten mit größeren Werten kürzer, wodurch Knoten näher aneinander stehen, und Kanten mit kleineren Werten werden länger.

### 5.19.540 chart.series.i[.].force.gravity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.1 ]

Der Schwerkraft-Faktor, der Knoten näher zum Mittelpunkt zieht. Die Knoten werden näher zum Mittelpunkt gezogen je größer der Wert ist.

### 5.19.541 chart.series.i[.].force.initLayout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das initiale Layout vor dem kräftebestimmten Layout, welches das Ergebnis des kräftebestimmten Layouts beeinflusst.

Setzt standardmäßig kein eigenes Layout und verwendet die Werte x und y der Knoten für deren Positionen. Existieren diese Werte nicht, wird die Position zufällig gesetzt.

Es kann auch das zirkuläre Layout 'circular' verwendet werden.

### 5.19.542 chart.series.i[.].force.layoutAnimation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Da das kräftebestimmte Layout erst nach mehreren Iterationen stabil ist, entscheidet dieser Parameter, ob die animierte Iteration des Layouts angezeigt werden soll. Es wird nicht empfohlen die Option in Browsern zu verwenden wenn es eine hohe Anzahl Knoten gibt (>100), da der Layout-Prozess den Browser zum Hängen bringen kann.

### 5.19.543 chart.series.i[].force.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen für kräftebestimmtes Layout. Das kräftebestimmte Layout simuliert ein Feder/Ladung-Modell, welches eine Abstoßung zwischen allen Knoten und eine Anziehung zwischen durch Kanten verbundenen Knoten definiert. Zuerst bewegen sich die Knoten in jeder Iteration unter den Einflüssen von Abstoßung und Anziehung. Nach mehreren Iterationen werden die Knoten in einer balancierten Position zur Ruhe kommen. So lässt sich das Modell des Graphens mit der niedrigsten lokalen Energie realisieren.

Das Ergebnis vom kräftebestimmten Layout hat gute Symmetrien und Häufungen, was auch ästhetisch ansprechend ist.

### 5.19.544 chart.series.i[].force.repulsion

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Der Abstoßungsfaktor zwischen Knoten. Die Abstoßung wird stärker und die Distanz zwischen 2 Knoten wird länger je größer dieser Wert wird.

Der Wert kann ein Array sein, um eine Spannweite an Abstoßungen darzustellen. In diesem Fall stehen höhere Werte für stärkere Abstoßung und niedrigere Werte für schwächere Abstoßung.

### 5.19.545 chart.series.i[].funnelAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Horizontale Ausrichtung. Standardmäßig zentral ausgerichtet.

Erlaubte Werte

- left
- right
- center

### 5.19.546 chart.series.i[].gap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Freiraum zwischen den Trapezen.

### 5.19.547 chart.series.i[].geoIndex

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index des geographischen Koordinatensystems mit dem dieses Diagramm kombiniert werden soll, nützlich für mehrere geografische Achsen in einem Diagramm.

### 5.19.548 chart.series.i[.].height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Höhe der Komponente.

### 5.19.549 chart.series.i[.].highlight.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'white' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.



### 5.19.550 chart.series.i[].highlight.itemStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.551 chart.series.i[].highlight.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 5.19.552 chart.series.i[].highlight.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.553 chart.series.i[].highlight.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Ringdiagramm-Sektoren.

Der Stil kann in `series.itemStyle` für Sektoren dieser Serie gesetzt werden, in `series.levels.itemStyle` für die Sektoren einer Ebene, und in `series.data.itemStyle` für einen einzelnen Sektor. Wenn `series.data.itemStyle` definiert ist, überdeckt es die Einstellungen von `series.itemStyle` und `series.levels.itemStyle`.

**Priorität: `series.data.itemStyle` > `series.levels.itemStyle` > `series.itemStyle`.**

In ECharts wird *emphasis* für Stile benutzt, wenn die Maus auf das Element zeigt. Bei Ringdiagrammen gibt es zwei zusätzliche States: *highlight*, um Elemente hervorzuheben, die mit dem ausgewählten in Verbindung stehen, und *downplay*, für Elemente, die nicht mit dem ausgewählten in Verbindung stehen.

### 5.19.554 chart.series.i[].highlight.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.555 chart.series.i[].highlight.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.556 chart.series.i[].highlight.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.557 chart.series.i[].highlight.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.558 chart.series.i[].highlight.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.559 chart.series.i[].highlight.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Ausrichtung des Textes. Man bemerke, dass 'left' für die innere Seite und 'right' für die äußere Seite steht.

#### Erlaubte Werte

- left
- center
- right

### 5.19.560 chart.series.i[].highlight.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.561 chart.series.i[].highlight.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 5.19.562 chart.series.i[].highlight.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.563 chart.series.i[].highlight.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 5.19.564 chart.series.i[].highlight.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.565 chart.series.i[].highlight.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.566 chart.series.i[].highlight.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.567 chart.series.i[].highlight.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.568 chart.series.i[.highlight.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.569 chart.series.i[.highlight.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.570 chart.series.i[].highlight.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.571 chart.series.i[].highlight.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.572 chart.series.i[].highlight.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.573 chart.series.i[].highlight.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 5.19.574 chart.series.i[].highlight.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

label setzt den Textstil für alle Sektoren.

**Priorität: series.data.label > series.levels.label > series.label**

Textbeschriftung der Sektoren, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 5.19.575 chart.series.i[].highlight.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.576 chart.series.i[].highlight.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.577 chart.series.i[].highlight.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.578 chart.series.i[].highlight.label.rich.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.579 chart.series.i[].highlight.label.rotate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'radial' ]

Ist der Wert eine Zahl, steht er für eine Rotation, von -90 bis 90 Grad, wobei positive Werte gegen den Uhrzeigersinn rotieren.

Außerdem kann der String 'radial', für radielle Rotation, oder 'tangential', für tangentielle Rotation, verwendet werden.

Der Wert ist standardmäßig 'radial'. Ist keine Rotation erwünscht, kann er auf 0 gesetzt werden.

#### Erlaubte Werte

- {integer}
- radial
- tangential

### 5.19.580 chart.series.i[.].highlight.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.581 chart.series.i[.].highlight.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.582 chart.series.i[].highlight.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.583 chart.series.i[].highlight.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.584 chart.series.i[].highlight.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 5.19.585 chart.series.i[].highlight.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.586 chart.series.i[].highlight.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 5.19.587 chart.series.i[].highlight.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.19.588 chart.series.i[].highlight.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 5.19.589 chart.series.i[].highlight.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.590 chart.series.i[].highlight.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.591 chart.series.i[].highlight.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.19.592 chart.series.i[].highlight.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.593 chart.series.i[].highlight.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil wenn die Maus auf in Verbindung stehende Objekte zeigt. Siehe highlightPolicy.

### 5.19.594 chart.series.i[].highlightPolicy

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'descendant' ]

Wenn die Maus auf einen Sektor zeigt, erhält dieser Sektor den *emphasis*-Stil. **Ist highlightPolicy auf 'descendant' gesetzt, erhält dieser Sektor und seine Nachfahren den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'ancestor' gesetzt, erhält dieser Sektor und seine Vorfahren den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'self' gesetzt, erhält der Sektor den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'none' gesetzt, erhalten andere Sektoren nicht den downplay-Stil.**

Es wird dispatchAction verwendet um bestimmte Sektoren hervorzuheben. Der ausgewählte Sektor verwendet den Stil *emphasis*, nach der gewählten highlightPolicy zusammenhängende Sektoren verwenden den Stil *highlight*, und andere verwenden den Stil *downplay*.

```

itemStyle: {
  color: 'yellow',
  borderWidth: 2,
  emphasis: {
    color: 'red'
  },
  highlight: {
    color: 'orange'
  },
  downplay: {
    color: '#ccc'
  }
}

```

#### Erlaubte Werte

- descendant
- ancestor
- self
- none

### 5.19.595 chart.series.i[].hoverAnimation

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Animationseffekt aktiviert werden soll wenn die Maus auf die Komponente zeigt.

### 5.19.596 chart.series.i[.hoverAnimation.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 5.19.597 chart.series.i[.hoverAnimation.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Verzögerung bevor die Animation beginnt. Eine Callback-Funktion kann verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
  return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
  return (params.count - 1 - params.index) * 30;
}
```

### 5.19.598 chart.series.i[].hoverAnimation.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Verzögerung bevor die Animation aktualisiert wird. Eine Callback-Funktion kann verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
  return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
  return (params.count - 1 - params.index) * 30;
}
```

### 5.19.599 chart.series.i[].hoverAnimation.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.600 chart.series.i[.].hoverAnimation.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.601 chart.series.i[.].hoverAnimation.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.602 chart.series.i[.].hoverAnimation.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.603 chart.series.i[].hoverAnimation.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 5.19.604 chart.series.i[.].hoverAnimation.object

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Animation beim Draufzeigen aktiviert sein soll.

Dieses Attribut kann auf der Wurzelebene einer Serie gesetzt werden, wodurch alle Datenelemente der Serie davon beeinflusst werden. Das Attribut kann auch bei jedem Eintrag in series-pictorialBar.data gesetzt werden, wodurch nur dieser Eintrag von dem Attribut beeinflusst wird.

Zum Beispiel:

```

series: [{
  hoverAnimation: ... // Beeinflusst alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    hoverAnimation: ... // Beeinflusst nur dieses Datenelement.
  }, {
    value: 56
    hoverAnimation: ... // Beeinflusst nur dieses Datenelement.
  }]
}]

```

### 5.19.605 chart.series.i[.].hoverOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Offset-Distanz des Sektors auf den gezeigt wird.

### 5.19.606 chart.series.i[.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.19.607 chart.series.i[.inactiveOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.05 ]

Bei der Verwendung von Brush-Auswahl werden nicht-ausgewählte Linien auf diese Transparenzrate gesetzt (was diese verdunkeln könnte).

### 5.19.608 chart.series.i[].initialTreeDepth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Die anfängliche Ebene (Tiefe) des Baums. Der Wurzelknoten ist Ebene 0, dann Ebene 1, 2, ..., bis zu den Blätterknoten. Diese Einstellung wird hauptsächlich zusammen mit Aus- und Einklappinteraktionen verwendet. Der Sinn ist, dass Knoten sich nicht gegenseitig überdecken sollen. Wird der Wert auf -1, null oder undefined gesetzt, werden alle Knoten ausgeklappt.

### 5.19.609 chart.series.i[].itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.610 chart.series.i[].itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

## 5.19.611 chart.series.i[.].itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 5.19.612 chart.series.i[.].itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.613 chart.series.i[[]].itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Grafikstil der Elemente. emphasis ist der Stil wenn das Element hervorgehoben ist, zum Beispiel wenn die Maus darauf zeigt oder über die Legendenverbindung.

#### 5.19.614 chart.series.i[.].itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.19.615 chart.series.i[.].itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.616 chart.series.i[].itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.617 chart.series.i[].itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.618 chart.series.i[].itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.619 chart.series.i[].label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.620 chart.series.i[].label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.621 chart.series.i[].label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.622 chart.series.i[].label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.623 chart.series.i[].label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.624 chart.series.i[].label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.625 chart.series.i[].label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.626 chart.series.i[].label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.627 chart.series.i[].label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.628 chart.series.i[].label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.629 chart.series.i[].label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.630 chart.series.i[].label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:



```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.631 chart.series.i[.].label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.632 chart.series.i[.].label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.633 chart.series.i[].label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 5.19.634 chart.series.i[].label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textbeschriftung der Komponenten, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 5.19.635 chart.series.i[].label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.636 chart.series.i[].label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.637 chart.series.i[].label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.638 chart.series.i[].label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.19.639 chart.series.i[.].label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.640 chart.series.i[.].label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.641 chart.series.i[.].label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.642 chart.series.i[.].label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.643 chart.series.i[.].label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.644 chart.series.i[.].label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.645 chart.series.i[.].label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.646 chart.series.i[].label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.647 chart.series.i[].label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Schattenverwischung des Textes selbst.

### 5.19.648 chart.series.i[].label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.649 chart.series.i[].label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.650 chart.series.i[].label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.19.651 chart.series.i[].label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

## 5.19.652 chart.series.i[].label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.653 chart.series.i[].labelLine.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.654 chart.series.i[.].labelLine.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenlinie.

### 5.19.655 chart.series.i[].labelLine.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.656 chart.series.i[].labelLine.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.657 chart.series.i[].labelLine.emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.658 chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.659 chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.660 chart.series.i[.].labelLine.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.661 chart.series.i[.].labelLine.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.662 chart.series.i[.].labelLine.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie im hervorgehobenen Status.

### 5.19.663 `chart.series.i[].labelLine.emphasis.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die hervorgehobene visuelle Hilfslinie angezeigt werden soll.

### 5.19.664 `chart.series.i[].labelLine.length`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des ersten Segments der visuellen Hilfslinie.

### 5.19.665 `chart.series.i[].labelLine.length2`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des zweiten Segments der visuellen Hilfslinie.

### 5.19.666 chart.series.i[.].labelLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.667 chart.series.i[.].labelLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Linie der visuellen Hilfslinie.

### 5.19.668 chart.series.i[].labelLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.669 chart.series.i[].labelLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.670 chart.series.i[].labelLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.671 chart.series.i[].labelLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.672 chart.series.i[].labelLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.673 chart.series.i[].labelLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.674 chart.series.i[].labelLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.675 chart.series.i[].labelLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie. Wird angezeigt wenn label.position auf 'outside' steht.

### 5.19.676 chart.series.i[].labelLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die visuelle Hilfslinie angezeigt werden soll.

### 5.19.677 chart.series.i[].labelLine.smooth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die visuelle Hilfslinie geglättet werden soll. Standardmäßig false und kann auf true oder Werte von 0 bis 1 gesetzt werden, welche den Glättungsgrad festlegen.

### 5.19.678 chart.series.i[].large

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Optimierung für große Datenmengen aktiviert werden soll. Kann aktiviert werden, wenn die Anzahl der Daten Performanzprobleme verursacht.

Nach der Aktivierung kann `largeThreshold` verwendet werden, um die Minimalzahl Daten zu bestimmen, für die die Optimierung verwendet wird.

Ist die Optimierung jedoch aktiviert, können einzelne Datensätze nicht mehr angepasst werden.

### 5.19.679 `chart.series.i[].largeThreshold`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

Die Grenze an Datensätzen, ab der die Optimierung verwendet wird.

### 5.19.680 `chart.series.i[].layout`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Layout der Komponente.

Mögliche Werte:

Tree:

- 'orthogonal': Orthogonales Layout annehmen. Knoten können links nach rechts sein, rechts nach links sein, etc.
- 'radial': Radiales Layout annehmen. Der Wurzelknoten ist in der Mitte, die anderen Ebenen an Knoten bilden Ringe um den Wurzelknoten.

Graph:

- 'none': Kein Layout annehmen. Verwende x, y der Knoten als Positionsangaben.
- 'circular': Kreislayout annehmen.
- 'force': Kräftebestimmtes Layout annehmen. Siehe graph.force für weitere Informationen.

Other:

- 'horizontal': Horizontales Layout annehmen.
- 'vertical': Vertikales Layout annehmen.

### 5.19.681 chart.series.i[].layoutIterations

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 32 ]

Die Iterationen über dem Layout, welche dazu benutzt werden, die Anzahl an Überlappungen im Graphen durch bessere Platzierung der Knoten zu verringern.

Standardmäßig 32 Iterationen.

Der Test zeigt, dass die Iterationen über dem Layout nicht weniger als der Default-Wert sein sollten.

### 5.19.682 chart.series.i[].leafDepth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Wenn leafDepth gesetzt wird, wird das "drill down"-Feature aktiviert, was bedeutet, dass beim Klicken auf einen Knoten im Baum, dieser Knoten als neue Wurzel gesetzt wird und seine Kinder angezeigt werden.

leafDepth steht für die Anzahl der Ebenen, die maximal auf einmal angezeigt werden. Steht leafDepth also auf 1, wird nur eine Ebene angezeigt.

leafDepth ist standardmäßig null/undefined, was bedeutet, dass "drill down" deaktiviert ist.

### 5.19.683 chart.series.i[].leaves.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.684 chart.series.i[].leaves.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

## Erlaubte Werte

- dashed
- dotted
- solid

5.19.685 `chart.series.i[].leaves.emphasis.itemStyle.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

5.19.686 `chart.series.i[].leaves.emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel `'rgb(128, 128, 128)'`. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel `'rgba(128, 128, 128, 0.5)'`. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel `'#ccc'`. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.687 chart.series.i[.].leaves.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Blätter.

### 5.19.688 chart.series.i[].leaves.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.689 chart.series.i[].leaves.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.690 chart.series.i[].leaves.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.691 chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.692 chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.693 chart.series.i[].leaves.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.694 chart.series.i[].leaves.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.695 chart.series.i[].leaves.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.696 chart.series.i[].leaves.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.697 chart.series.i[].leaves.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.698 chart.series.i[].leaves.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.699 chart.series.i[].leaves.emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.700 chart.series.i[.].leaves.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.701 chart.series.i[.].leaves.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.702 chart.series.i[.].leaves.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.703 chart.series.i[.].leaves.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.704 chart.series.i[.].leaves.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.705 chart.series.i[.].leaves.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.19.706 chart.series.i[].leaves.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Blattbeschriftungen.

### 5.19.707 chart.series.i[].leaves.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.708 chart.series.i[].leaves.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.709 chart.series.i[.].leaves.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom

- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.710 chart.series.i[.].leaves.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.711 chart.series.i[.].leaves.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.712 chart.series.i[].leaves.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.713 chart.series.i[].leaves.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.714 chart.series.i[].leaves.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.715 chart.series.i[].leaves.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.716 chart.series.i[].leaves.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.717 chart.series.i[].leaves.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.718 chart.series.i[].leaves.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.719 chart.series.i[].leaves.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.720 chart.series.i[].leaves.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.721 chart.series.i[].leaves.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.722 chart.series.i[].leaves.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.723 chart.series.i[].leaves.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.724 chart.series.i[].leaves.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.725 chart.series.i[].leaves.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Komponenten.

### 5.19.726 chart.series.i[].leaves.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.727 chart.series.i[].leaves.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.728 chart.series.i[.].leaves.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1.5 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.729 chart.series.i[.].leaves.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.730 chart.series.i[].leaves.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil des Blattknotens im Baum.

### 5.19.731 chart.series.i[].leaves.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.732 chart.series.i[].leaves.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.



Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.733 chart.series.i[].leaves.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.734 chart.series.i[].leaves.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.735 chart.series.i[].leaves.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.736 chart.series.i[.].leaves.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.737 chart.series.i[.].leaves.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.738 chart.series.i[].leaves.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 5.19.739 chart.series.i[].leaves.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.740 chart.series.i[].leaves.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 5.19.741 chart.series.i[].leaves.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.742 chart.series.i[].leaves.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.743 chart.series.i[].leaves.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.744 chart.series.i[].leaves.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.745 chart.series.i[.].leaves.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.746 chart.series.i[.].leaves.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.747 chart.series.i[].leaves.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.748 chart.series.i[].leaves.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.749 chart.series.i[].leaves.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschreibt den Stil der zu den Blattknoten gehörenden Beschriftung.

### 5.19.750 chart.series.i[].leaves.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.751 chart.series.i[].leaves.label.padding

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.752 chart.series.i[.].leaves.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left

- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.753 chart.series.i[.].leaves.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.754 chart.series.i[.].leaves.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.755 chart.series.i[].leaves.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.756 chart.series.i[].leaves.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.757 chart.series.i[].leaves.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.758 chart.series.i[].leaves.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.759 chart.series.i[].leaves.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.760 chart.series.i[].leaves.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.761 chart.series.i[].leaves.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.762 chart.series.i[].leaves.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.763 chart.series.i[].leaves.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.764 chart.series.i[].leaves.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.765 chart.series.i[].leaves.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.19.766 chart.series.i[].leaves.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 5.19.767 chart.series.i[].leaves.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.768 chart.series.i[].leaves.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen speziell für die Blattknoten. Zum Beispiel für andere Positionen der Beschriftungen von Blattknoten und inneren Knoten.

### 5.19.769 chart.series.i[].left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 5.19.770 chart.series.i[].legendHoverLink

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Diagramm hervorgehoben werden soll, wenn die Maus auf die Legende zeigt.

### 5.19.771 chart.series.i[].levels.childrenVisibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Kinder werden nicht angezeigt wenn die Größe des Bereichs eines Knotens kleiner als dieser Wert ist (Einheit: Quadratpixel).

Durch diese Einstellung können die Details von Knoten versteckt werden, wenn der rechteckige Bereich nicht groß genug ist. Wenn Nutzer auf den Knoten hereinzoomen, werden die Kindknoten angezeigt wenn der Bereich größer als der angegebene Wert wird.

Siehe series-treemap.levels für Details über visuelles Encoding.

Tip: In treemap kann das childrenVisibleMin an mehreren Stellen auftauchen:

- In series-treemap, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.772 `chart.series.i[].levels.colorAlpha`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Transparenzraten (Alpha-Kanal) der Serie an. Akzeptiert Werte von 0 bis 1.

Der Wert kann zum Beispiel [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorAlpha` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.773 `chart.series.i[].levels.colorMappingBy`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'index' ]

Spezifiziert die Regel, nach der jeder Knoten seine Farbe aus der Farbliste erhält. Mögliche Werte:

- 'value':

Mappt `series-treemap.data.value` auf die Farben.

So zeigen die Farben der Knoten ihre Werte an.

`visualDimension` kann verwendet werden, um zu bestimmen, welche Datendimension für das visuelle Mapping verwendet wird.

- 'index':

Mappt den Index (Ordinalzahl) des Knotens auf die Farben. Dadurch bekommt in einer Ebene der erste Knoten die erste Farbe der Liste und der zweite Knoten die zweite Farbe.

So können aufeinanderfolgende Knoten durch ihre Farben unterschieden werden.

- 'id':

Mappt `series-treemap.data.id` auf die Farben.

Da die `id` verwendet wird um die Knoten zu identifizieren, behalten die Knoten so ihre ursprüngliche Farbe, wenn der Baum durch `setOption` verändert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

## Erlaubte Werte

- value
- index
- id

### 5.19.774 `chart.series.i[.].levels.colorSaturation`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Sättigungswerten der Serie an. Akzeptiert Werte von 0 bis 1.

Zum Beispiel kann `colorSaturation` [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 5.19.775 `chart.series.i[].levels.emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbe eines Knotens. Verwendet standardmäßig die globale Palette `option.color`.

### 5.19.776 `chart.series.i[].levels.emphasis.itemStyle.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tips: In treemap kann das `ItemStyle` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.777 `chart.series.i[].levels.emphasis.label.align`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

#### 5.19.778 chart.series.i[.].levels.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.779 chart.series.i[.].levels.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.780 chart.series.i[.].levels.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.781 `chart.series.i[.].levels.emphasis.label.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.782 `chart.series.i[.].levels.emphasis.label.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.783 `chart.series.i[.].levels.emphasis.label.distance`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.784 chart.series.i[].levels.emphasis.label.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 5.19.785 chart.series.i[].levels.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.786 chart.series.i[].levels.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.787 chart.series.i[.].levels.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.788 chart.series.i[.].levels.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.789 chart.series.i[.].levels.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.790 chart.series.i[.].levels.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.791 chart.series.i[].levels.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.792 chart.series.i[].levels.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der auf der Ebene hervorgehobenen Knoten.

### 5.19.793 chart.series.i[].levels.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.794 chart.series.i[].levels.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.795 chart.series.i[].levels.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

## Position der Beschriftung

### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

## 5.19.796 chart.series.i[.].levels.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.797 chart.series.i[.].levels.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.



## 5.19.798 chart.series.i[].levels.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.799 chart.series.i[].levels.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.800 chart.series.i[].levels.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.801 chart.series.i[].levels.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.802 chart.series.i[].levels.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.803 chart.series.i[].levels.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.804 chart.series.i[].levels.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.805 chart.series.i[].levels.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.806 chart.series.i[].levels.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.807 chart.series.i[.].levels.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.808 chart.series.i[.].levels.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.809 chart.series.i[].levels.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.810 chart.series.i[].levels.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.811 chart.series.i[].levels.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Beschriftungen der hervorgehobenen Knoten der Ebene.

### 5.19.812 chart.series.i[].levels.emphasis.upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.19.813 chart.series.i[.].levels.emphasis.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.814 chart.series.i[].levels.emphasis.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.815 chart.series.i[].levels.emphasis.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.816 chart.series.i[].levels.emphasis.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.817 chart.series.i[.].levels.emphasis.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.818 chart.series.i[.].levels.emphasis.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 5.19.819 chart.series.i[].levels.emphasis.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 5.19.820 chart.series.i[].levels.emphasis.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.821 chart.series.i[].levels.emphasis.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 12 ]

Schriftgröße.

## 5.19.822 chart.series.i[.].levels.emphasis.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.19.823 chart.series.i[.].levels.emphasis.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.824 chart.series.i[.].levels.emphasis.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.825 chart.series.i[.].levels.emphasis.upperLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.826 chart.series.i[.].levels.emphasis.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.827 chart.series.i[].levels.emphasis.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Oberbeschriftung des hervorgehobenen Knotens.

### 5.19.828 chart.series.i[].levels.emphasis.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 5.19.829 chart.series.i[.].levels.emphasis.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.830 chart.series.i[.].levels.emphasis.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.831 chart.series.i[.].levels.emphasis.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.832 chart.series.i[.].levels.emphasis.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 5.19.833 chart.series.i[].levels.emphasis.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.834 chart.series.i[].levels.emphasis.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.835 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.836 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.837 chart.series.i[].levels.emphasis.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.838 chart.series.i[].levels.emphasis.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.839 chart.series.i[].levels.emphasis.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.840 chart.series.i[].levels.emphasis.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.841 chart.series.i[].levels.emphasis.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.842 chart.series.i[.].levels.emphasis.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.843 chart.series.i[.].levels.emphasis.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.844 chart.series.i[].levels.emphasis.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.845 chart.series.i[].levels.emphasis.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.846 chart.series.i[.].levels.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Die Rahmenfarbe und Lückenfarbe eines Knotens.

### 5.19.847 chart.series.i[.].levels.itemStyle.borderColorSaturation

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbsättigung eines Rahmens oder Zwischenraums. Die Werte gehen von 0 bis 1.

Tips:

Wenn borderColorSaturation gesetzt ist wird borderColor deaktiviert und die endgültige Rahmenfarbe stattdessen dadurch berechnet, dass die Farbe dieses Knotens (die entweder explizit definiert oder vom Elternknoten geerbt ist) mit borderColorSaturation gemischt wird.

So kann ein Effekt implementiert werden: Verschiedene Sektionen haben jeweils andere Farbtöne in ihren Zwischenräumen, wodurch Nutzer sowohl Sektionen als auch Ebenen leicht unterscheiden können.

### Wie man Verwirrung beim Setzen des Knotenrahmens vermeidet

Wenn alle Rahmen/Zwischenräume auf die selbe Farbe gesetzt sind, kann Verwirrung auftreten, wenn Rechtecke mehrerer Ebenen auf einmal dargestellt werden.

Stattdessen kann man die Farbe des Zwischenraums setzen und die Sättigung dann mit `borderColorSaturation` verändern.

## 5.19.848 `chart.series.i[].levels.itemStyle.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite eines Knotens. Ist der Wert auf 0 gesetzt, gibt es keinen Rahmen.

Hinweis: Abstände zwischen Kindknoten werden durch `gapWidth` spezifiziert.

## 5.19.849 `chart.series.i[].levels.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbe eines Knotens. Verwendet standardmäßig die globale Palette `option.color`.

## 5.19.850 `chart.series.i[].levels.itemStyle.colorAlpha`

<b>Datentyp</b>	DECIMAL
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Transparenzrate eines Knotens, welche zwischen 0 und 1 liegt.

### 5.19.851 chart.series.i[.].levels.itemStyle.colorSaturation

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbsättigung eines Knotens. Die Werte gehen von 0 bis 1.

### 5.19.852 chart.series.i[.].levels.itemStyle.gapWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Freiraum zwischen Kindknoten.

### 5.19.853 chart.series.i[].levels.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tips: In treemap kann das ItemStyle Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.854 chart.series.i[].levels.itemStyle.strokeColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Strichfarbe jedes Rechtecks.

### 5.19.855 chart.series.i[].levels.itemStyle.strokeWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Strichbreite jedes Rechtecks.

## 5.19.856 chart.series.i[.].levels.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 5.19.857 chart.series.i[.].levels.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.858 chart.series.i[.].levels.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.859 chart.series.i[.].levels.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.860 chart.series.i[.].levels.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.861 chart.series.i[.].levels.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.862 chart.series.i[.].levels.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.863 chart.series.i[.].levels.label.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 5.19.864 chart.series.i[.].levels.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.865 chart.series.i[.].levels.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.866 chart.series.i[.].levels.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### 5.19.867 chart.series.i[.].levels.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.868 chart.series.i[.].levels.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:



```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.869 chart.series.i[].levels.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.870 chart.series.i[].levels.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.871 chart.series.i[].levels.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Beschriftung der Knoten

Tips: In treemap kann das label Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.872 chart.series.i[].levels.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.873 chart.series.i[].levels.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.874 chart.series.i[].levels.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.875 chart.series.i[.].levels.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.876 chart.series.i[].levels.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 5.19.877 chart.series.i[].levels.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.878 chart.series.i[].levels.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.879 chart.series.i[].levels.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.880 chart.series.i[].levels.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.881 chart.series.i[].levels.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.882 chart.series.i[].levels.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.883 chart.series.i[].levels.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.884 chart.series.i[].levels.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.885 chart.series.i[].levels.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.886 chart.series.i[.].levels.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.887 chart.series.i[.].levels.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.888 chart.series.i[].levels.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.889 chart.series.i[].levels.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.890 chart.series.i[].levels.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

#### Konfiguration auf mehreren Ebenen

treemap verwendet eine vierstufige Konfiguration:

"jeder Knoten" --> "jede Ebene" --> "jede Serie".

Wir können also jeden Knoten konfigurieren, jede Ebene des Baums konfigurieren, oder Konfigurationen für die gesamte Serie setzen. Die höchste Priorität hat die Konfiguration des Knotens.

levels ist die Konfiguration der einzelnen Ebenen, was am am meisten verwendet wird.

Zum Beispiel:

```
// Bemerke dass die Datenstruktur kein Baum ist, sondern ein Wald
data: [
  {
    name: 'nodeA',
    children: [
      {name: 'nodeAA'},
      {name: 'nodeAB'},
    ]
  },
  {
    name: 'nodeB',
    children: [
      {name: 'nodeBA'}
    ]
  }
],
levels: [
  {...}, // Konfigurationen auf der obersten Ebene der Struktur
         // (die Ebene die die obigen 'nodeA' und 'nodeB' enthält).
  {...}, // Konfigurationen der nächsten Ebene
         // (die Ebene die die obigen 'nodeAA', 'node AB', 'nodeBA' enthält)
  {...}, // Konfigurationen der nächsten Ebene
  ...
]
```

### Die Regeln über visuelles Mapping

Wenn man eine treemap entwickelt liegt der Hauptfokus darauf, wie man "verschiedene Ebenen" und "verschiedene Kategorien der selben Ebene" visuell unterscheidbar macht, was passende Werte der Rechtecksfarben, Rahmenbreiten, Rahmenfarben und auch Farbsättigungen auf allen Ebenen benötigt.

treemap verwendet diese Regel für die visuelle Konfiguration: Jede Ebene konfiguriert seinen visuellen Wert basierend auf den Einstellungen (color, colorSaturation, borderColor, colorSaturation) auf dieser Ebene. Liegt keine explizite Konfiguration vor, wird die Konfiguration des Elternknotens geerbt.

So kann dieser Effekt konfiguriert werden: Setze eine Farbliste auf der Elternebene und setze eine Farbsättigung auf der Kindebene. Jeder Knoten auf der Elternebene kriegt dann eine Farbe aus der Farbliste, und jeder Knoten auf der Kindebene kriegt eine Farbsättigung und kombiniert sie mit der Farbe seines Elternknotens um seine endgültige Farbe zu erhalten.

### Dimensionen und "Extra visuelles Mapping"

Siehe das untere Beispiel: jedes value-Feld ist ein Array, in dem jeder Eintrag eine Dimension repräsentiert.

```
[
  {
    value: [434, 6969, 8382],
    children: [
      {
        value: [1212, 4943, 5453],
        id: 'someid-1',
        name: 'Beschreibung dieses Knotens',
        children: [...]
      },
      {
        value: [4545, 192, 439],
        id: 'someid-2',
        name: 'Beschreibung dieses Knotens',
        children: [...]
      },
      ...
    ]
  },
  {
    value: [23, 59, 12],
    children: [...]
  },
  ...
]
```

treemap wird die erste Dimension (den ersten Eintrag des Arrays) auf "area" mappen. Falls wir mehr Informationen darstellen wollen, können wir eine weitere Dimension (spezifiziert durch `series-treemap.visualDimension`) auf einen anderen visuellen Typ, wie `colorSaturation`, mappen und so weiter.

### 5.19.891 `chart.series.i[[]].levels.upperLabel.align`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `align` im rich nicht gesetzt wird, wird das `align` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

## Erlaubte Werte

- left
- center
- right

### 5.19.892 chart.series.i[.].levels.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.893 chart.series.i[].levels.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.894 chart.series.i[].levels.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.895 chart.series.i[].levels.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.896 chart.series.i[.].levels.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.897 chart.series.i[.].levels.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').



## 5.19.898 chart.series.i[].levels.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

## 5.19.899 chart.series.i[].levels.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.19.900 chart.series.i[].levels.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.901 chart.series.i[.].levels.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 5.19.902 chart.series.i[.].levels.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.903 chart.series.i[.].levels.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

#### 5.19.904 chart.series.i[.].levels.upperLabel.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Hohe des Beschriftungsbereichs.

#### 5.19.905 chart.series.i[.].levels.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.906 chart.series.i[.].levels.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

upperLabel spezifiziert, ob die Beschriftung eines Knotens mit Kindknoten angezeigt werden soll. Wenn upperLabel.show auf true gesetzt ist, werden die Beschriftungen solcher Elternknoten angezeigt.

So wie bei series-treemap.label kann die Option upperLabel an der Wurzel von series-treemap gesetzt werden, oder in series-treemap.level, oder in jedem Eintrag von series-treemap.data.

Insbesondere spezifiziert series-treemap.label den Stil, wenn der Knoten ein Blatt ist, während series-treemap.upperLabel den Stil spezifiziert, wenn ein Knoten Kinder hat. In diesem Fall wird die Beschriftung im inneren oberen Bereich des Knotens angezeigt.

Tips: In treemap kann das upperLabel Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.907 chart.series.i[.].levels.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.908 chart.series.i[.].levels.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.909 chart.series.i[.].levels.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

## Position der Beschriftung

### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

## 5.19.910 chart.series.i[.].levels.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.911 chart.series.i[.].levels.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.



## 5.19.912 chart.series.i[].levels.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.913 chart.series.i[].levels.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.914 chart.series.i[].levels.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.915 chart.series.i[.].levels.upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.916 chart.series.i[.].levels.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.917 chart.series.i[.].levels.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.918 chart.series.i[.].levels.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.919 chart.series.i[.].levels.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.920 chart.series.i[.].levels.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.921 chart.series.i[.].levels.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.922 chart.series.i[.].levels.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.923 chart.series.i[].levels.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.924 chart.series.i[].levels.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.925 chart.series.i[].levels.visibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Ein Knoten wird nicht angezeigt wenn die Größe seines Bereichs kleiner als dieser Wert ist (Einheit: px quadrat).

So werden kleine Knoten, die ansonsten eng aneinander stehen würden, versteckt. Wenn ein Nutzer die treemap vergrößert, vergrößert sich auch die Bereichsgröße, und das Rechteck wird angezeigt, wenn sie über diese Grenze steigt.

Siehe serie-treemap.levels für Details zu visuellem Encoding.

Tips: In treemap kann das visibleMin Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.926 chart.series.i[].levels.visualDimension

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

treemap kann beliebige Dimensionen der Daten ins Visuelle mappen.

Der Wert von `series-treemap.data` kann ein Array sein. Jeder Eintrag des Arrays steht für eine "dimension". `visualDimension` spezifiziert die Dimension, auf der visuelles Mapping durchgeführt wird.

Siehe `series-treemap.levels` für Details zu visuellem Encoding.

Tips: In treemap kann das `visualDimension` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.927 `chart.series.i[.].levels.visualMax`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn `colorMappingBy` auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von `visualMin` oder `visualMax` bestimmt werden.

### 5.19.928 `chart.series.i[.].levels.visualMin`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Der minimale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn colorMappingBy auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von visualMin oder visualMax bestimmt werden.

### 5.19.929 chart.series.i[].lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.930 chart.series.i[.].lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Linienstil

## 5.19.931 chart.series.i[].lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 5.19.932 chart.series.i[].lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.933 chart.series.i[].lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.934 chart.series.i[].lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.935 chart.series.i[].lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.936 chart.series.i[.].lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.937 chart.series.i[.].lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Linienbreite.

### 5.19.938 chart.series.i[.].links.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

#### 5.19.939 chart.series.i[].links.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.940 chart.series.i[.links.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.941 chart.series.i[.links.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.942 chart.series.i[].links.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 5.19.943 chart.series.i[].links.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

## 5.19.944 chart.series.i[].links.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.945 chart.series.i[.links.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.946 chart.series.i[.links.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- oblique
- italic

### 5.19.947 chart.series.i[.links.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.19.948 chart.series.i[.].links.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.949 chart.series.i[.links.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.950 chart.series.i[.links.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.951 chart.series.i[.links.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Beschriftung des hervorgehobenen Elements.

### 5.19.952 chart.series.i[].links.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.953 chart.series.i[].links.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte  
position: [10, 10],  
// Relative Prozentwerte  
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.954 chart.series.i[.links.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.955 chart.series.i[.links.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.956 chart.series.i[].links.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.957 chart.series.i[].links.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.958 chart.series.i[].links.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.959 chart.series.i[].links.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.960 chart.series.i[].links.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.961 chart.series.i[].links.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.962 chart.series.i[].links.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.963 chart.series.i[].links.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.964 chart.series.i[].links.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.965 `chart.series.i[.links.emphasis.label.textShadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.966 `chart.series.i[.links.emphasis.label.verticalAlign`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im rich nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 5.19.967 chart.series.i[.links.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.968 chart.series.i[.links.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.969 chart.series.i[].links.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Liniestil des hervorgehobenen Elements.

### 5.19.970 chart.series.i[].links.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.971 chart.series.i[].links.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.972 chart.series.i[].links.emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.973 chart.series.i[].links.emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.974 chart.series.i[].links.emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.975 chart.series.i[.links.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.976 chart.series.i[.links.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.977 chart.series.i[].links.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des hervorgehobenen Elements.

### 5.19.978 chart.series.i[].links.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right



### 5.19.979 chart.series.i[.links.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.980 chart.series.i[.links.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

## 5.19.981 chart.series.i[.links.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.982 chart.series.i[.links.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 5.19.983 chart.series.i[.links.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.19.984 chart.series.i[].links.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.985 chart.series.i[].links.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.986 chart.series.i[].links.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.987 chart.series.i[.links.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.988 chart.series.i[.links.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.989 chart.series.i[.].links.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.990 chart.series.i[.].links.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.991 chart.series.i[].links.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Elements.

### 5.19.992 chart.series.i[].links.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.993 chart.series.i[.].links.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom



- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.994 chart.series.i[.links.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.995 chart.series.i[.links.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.996 chart.series.i[.links.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.997 chart.series.i[.links.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 5.19.998 chart.series.i[.links.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.999 chart.series.i[].links.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1000 chart.series.i[].links.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.1001 chart.series.i[].links.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1002 chart.series.i[].links.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1003 chart.series.i[].links.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1004 chart.series.i[].links.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1005 chart.series.i[.links.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1006 chart.series.i[.links.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

### Erlaubte Werte

- top
- middle
- bottom

### 5.19.1007 chart.series.i[].links.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1008 chart.series.i[].links.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```



### 5.19.1009 chart.series.i[].links.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Krümmung der Kante, unterstützt Werte von 0 bis 1. Die Krümmung wird stärker wenn der Wert größer wird.

### 5.19.1010 chart.series.i[].links.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Kanten.

### 5.19.1011 chart.series.i[].links.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1012 chart.series.i[].links.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1013 chart.series.i[].links.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1014 chart.series.i[].links.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1015 chart.series.i[.links.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1016 chart.series.i[.links.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

## 5.19.1017 chart.series.i[].links.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

## 5.19.1018 chart.series.i[].links.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beziehungsdaten zwischen Knoten. Beispiel:

```
links: [{
  source: 'n1',
  target: 'n2'
}, {
  source: 'n2',
  target: 'n3'
}]
```

## 5.19.1019 chart.series.i[].links.source

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Quellknotens der Kante.

### 5.19.1020 chart.series.i[].links.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Kantenenden. Kann ein Array mit zwei Elementen sein, um die Enden separat zu definieren, oder ein String, um sie zusammen zu definieren.

### 5.19.1021 chart.series.i[].links.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße der Kantenenden. Kann ein Array mit zwei Elementen sein, um die Enden separat zu definieren, oder ein String, um sie zusammen zu definieren.

### 5.19.1022 chart.series.i[].links.target

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Zielknotens der Kante.

### 5.19.1023 chart.series.i[.links.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kantenwert. Kann in kräftebasiertem Graphen auf die kantenlänge gemappt werden.

### 5.19.1024 chart.series.i[.markArea.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 5.19.1025 chart.series.i[].markArea.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
    // Verzögerung für spätere Daten ist größer  
    return idx * 100;  
}
```

### 5.19.1026 chart.series.i[].markArea.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 5.19.1027 chart.series.i[].markArea.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.1028 chart.series.i[].markArea.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:



```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.1029 chart.series.i[].markArea.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn

- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1030 chart.series.i[[]].markArea.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut

- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1031 chart.series.i[].markArea.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 5.19.1032 chart.series.i[].markArea.data.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Ausmaß des Bereichs wird in data festgelegt. data akzeptiert Arrays mit zwei Elementen, welche den linken oberen und den rechten unteren Punkt der rechteckigen Markierungsebene beschreiben. Jeder Eintrag kann wie folgt definiert werden:

1. Spezifiziere die Koordinaten im Koordinatensystem des Bildschirms mit x und y, wobei die Werte Pixelwerte (wie 5) oder Prozentwerte (wie '35%') sein können.
2. Spezifiziere die Koordinaten im (kartesischen) Koordinatensystem der Daten durch coord, welches auch die Werte 'min', 'max' und 'average' akzeptiert (z.B. coord: [23, 'min'] oder coord: ['average', 'max']).
3. Spezifiziere den minimalen, maximalen oder durchschnittlichen Wert von series.data durch type, wobei valueIndex oder valueDim verwendet werden können um die Dimension zu bestimmen, auf der die Werte berechnet werden.
4. In einem kartesischen System können nur xAxis oder yAxis spezifiziert werden, um einen markierten Bereich basierend auf nur der X- oder Y-Achse zu definieren

Die Priorität ist wie oben wenn mehr als eine der oberen Definitionen benutzt wird.

```
data: [
  [
    {
      name: 'Vom Durchschnitt zum Maximum',
      type: 'average'
    },
    {
      type: 'max'
    }
  ],
  [
    {
      name: 'Bereich zwischen zwei Punkten in Datenkoordinaten markieren',
      coord: [10, 20]
    },
    {
      coord: [20, 30]
    }
  ], [
    {
      name: 'Von 60 bis 80',
      yAxis: 60
    },
    {
      yAxis: 80
    }
  ], [
    {
      name: 'Markierter Bereich umschließt alle Daten'
      coord: ['min', 'min']
    },
    {
      coord: ['max', 'max']
    }
  ], [
    {
      name: 'Bereich um zwei Bildschirmpunkte markieren',
      x: 100,
      y: 100
    }, {
      x: '90%',
      y: '10%'
    }
  ]
]
```

## 5.19.1033 chart.series.i[].markArea.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.19.1034 chart.series.i[].markArea.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

## 5.19.1035 chart.series.i[].markArea.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.1036 chart.series.i[].markArea.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1037 chart.series.i[].markArea.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.19.1038 chart.series.i[].markArea.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

## 5.19.1039 chart.series.i[].markArea.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.



### 5.19.1040 chart.series.i[].markArea.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1041 chart.series.i[].markArea.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil des hervorgehobenen Markierungsbereichs.

### 5.19.1042 chart.series.i[].markArea.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1043 chart.series.i[].markArea.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1044 chart.series.i[].markArea.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1045 chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1046 chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1047 chart.series.i[].markArea.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Markierungsbereichs.

### 5.19.1048 chart.series.i[].markArea.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1049 chart.series.i[].markArea.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1050 chart.series.i[].markArea.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1051 chart.series.i[].markArea.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1052 chart.series.i[].markArea.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1053 chart.series.i[].markArea.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.1054 chart.series.i[].markArea.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1055 chart.series.i[].markArea.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]



Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1056 chart.series.i[].markArea.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.1057 chart.series.i[].markArea.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1058 chart.series.i[].markArea.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1059 chart.series.i[].markArea.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.1060 chart.series.i[].markArea.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 5.19.1061 chart.series.i[].markArea.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1062 chart.series.i[].markArea.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1063 chart.series.i[].markArea.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.1064 chart.series.i[].markArea.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1065 chart.series.i[].markArea.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1066 chart.series.i[].markArea.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.1067 chart.series.i[].markArea.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1068 chart.series.i[].markArea.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.1069 chart.series.i[].markArea.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

## Erlaubte Werte

- normal
- italic
- oblique

## 5.19.1070 chart.series.i[].markArea.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.1071 chart.series.i[].markArea.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.19.1072 chart.series.i[].markArea.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 5.19.1073 chart.series.i[].markArea.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Beschriftung des hervorgehobenen Markierungsbereichs.

### 5.19.1074 chart.series.i[].markArea.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.1075 chart.series.i[].markArea.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.1076 chart.series.i[].markArea.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.1077 chart.series.i[.].markArea.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

## Schlagworte / Tags

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.1078 chart.series.i[].markArea.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.1079 chart.series.i[].markArea.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.1080 chart.series.i[].markArea.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.1081 chart.series.i[].markArea.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1082 chart.series.i[].markArea.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1083 chart.series.i[].markArea.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1084 chart.series.i[].markArea.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1085 chart.series.i[].markArea.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1086 chart.series.i[].markArea.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.19.1087 chart.series.i[].markArea.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 5.19.1088 chart.series.i[].markArea.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 5.19.1089 chart.series.i[].markArea.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1090 chart.series.i[].markArea.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.1091 chart.series.i[].markArea.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.19.1092 chart.series.i[].markArea.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.19.1093 chart.series.i[].markArea.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 5.19.1094 chart.series.i[].markArea.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 5.19.1095 chart.series.i[].markArea.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.1096 chart.series.i[].markArea.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1097 chart.series.i[].markArea.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.19.1098 chart.series.i[].markArea.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Markierungsbereichs.

### 5.19.1099 chart.series.i[].markArea.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.1100 chart.series.i[].markArea.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.1101 chart.series.i[].markArea.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom

- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.1102 chart.series.i[].markArea.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.1103 chart.series.i[].markArea.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 5.19.1104 chart.series.i[].markArea.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.1105 chart.series.i[].markArea.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.1106 chart.series.i[].markArea.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1107 chart.series.i[].markArea.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1108 chart.series.i[].markArea.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1109 chart.series.i[].markArea.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1110 chart.series.i[].markArea.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1111 chart.series.i[].markArea.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1112 chart.series.i[].markArea.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1113 chart.series.i[].markArea.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1114 chart.series.i[].markArea.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1115 chart.series.i[].markArea.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.1116 chart.series.i[].markArea.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1117 chart.series.i[].markArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wird verwendet um einen Bereich im Diagramm zu markieren. Zum Beispiel ein Zeitintervall.

### 5.19.1118 chart.series.i[].markArea.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

### 5.19.1119 chart.series.i[].markLine.animation

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 5.19.1120 chart.series.i[].markLine.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
  // Verzögerung für spätere Daten ist größer  
  return idx * 100;  
}
```

### 5.19.1121 chart.series.i[].markLine.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
    // Verzögerung für spätere Daten ist größer  
    return idx * 100;  
}
```

### 5.19.1122 chart.series.i[].markLine.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {  
    // Dauer ist für spätere Daten länger  
    return idx * 100;  
}
```

### 5.19.1123 chart.series.i[].markLine.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.1124 chart.series.i[].markLine.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut



- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1125 chart.series.i[].markLine.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1126 chart.series.i[].markLine.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 5.19.1127 chart.series.i[].markLine.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagnworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 5.19.1128 chart.series.i[].markLine.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der hervorgehobenen Markierungslinie.

## 5.19.1129 chart.series.i[].markLine.label.emphasis.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'end' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

## 5.19.1130 chart.series.i[].markLine.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1131 chart.series.i[].markLine.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Seriename.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.1132 chart.series.i[].markLine.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der Markierungslinie.

### 5.19.1133 chart.series.i[].markLine.label.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'end' ]

Position der Beschriftung.

### Erlaubte Werte

- start
- middle
- end

## 5.19.1134 chart.series.i[].markLine.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

## 5.19.1135 chart.series.i[].markLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1136 chart.series.i[].markLine.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Krümmung der Kante, welche Werte von 0 bis 1 unterstützt. Je größer der Wert, desto größer die Krümmung.

### 5.19.1137 chart.series.i[].markLine.lineStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1138 chart.series.i[].markLine.lineStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Markierungslinie.

### 5.19.1139 chart.series.i[].markLine.lineStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1140 chart.series.i[].markLine.lineStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 5.19.1141 chart.series.i[].markLine.lineStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 5.19.1142 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.19.1143 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1144 chart.series.i[].markLine.lineStyle.emphasis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.1145 chart.series.i[].markLine.lineStyle.emphasis.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 5.19.1146 chart.series.i[].markLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Markierungslinie.

### 5.19.1147 chart.series.i[].markLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1148 chart.series.i[].markLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1149 chart.series.i[].markLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1150 chart.series.i[].markLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1151 chart.series.i[].markLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1152 chart.series.i[].markLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.1153 chart.series.i[].markLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.



### 5.19.1154 chart.series.i[].markLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verwende zu Illustrationszwecken eine Linie im Graphen.

### 5.19.1155 chart.series.i[].markLine.precision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Präzision des Wertes der Markierungslinie, was nützlich ist, wenn die Linie den durchschnittlichen Wert anzeigen soll.

### 5.19.1156 chart.series.i[].markLine.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

### 5.19.1157 chart.series.i[].markLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symboltyp an den beiden Enden der Markierungslinie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren. Siehe data.symbol für weitere Informationen zum Format.

### 5.19.1158 chart.series.i[].markLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße an den beiden Enden der Markierungslinie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren.

**Vorsicht:** Es können nicht Breite und Höhe getrennt gesetzt werden wie beim normalen SymbolSize.

### 5.19.1159 chart.series.i[].markPoint.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 5.19.1160 chart.series.i[].markPoint.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 5.19.1161 chart.series.i[].markPoint.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 5.19.1162 chart.series.i[].markPoint.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.1163 chart.series.i[].markPoint.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 5.19.1164 chart.series.i[].markPoint.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut

- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1165 chart.series.i[.].markPoint.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn

- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.19.1166 chart.series.i[].markPoint.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 5.19.1167 chart.series.i[].markPoint.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.1168 chart.series.i[].markPoint.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.1169 chart.series.i[].markPoint.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.1170 chart.series.i[].markPoint.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1171 chart.series.i[].markPoint.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.19.1172 chart.series.i[].markPoint.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 5.19.1173 chart.series.i[].markPoint.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.19.1174 chart.series.i[].markPoint.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.19.1175 chart.series.i[].markPoint.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil des hervorgehobenen Markierungspunktes.

### 5.19.1176 chart.series.i[].markPoint.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1177 chart.series.i[].markPoint.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 5.19.1178 chart.series.i[].markPoint.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 5.19.1179 chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.19.1180 chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1181 chart.series.i[].markPoint.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Markierungspunktes.

### 5.19.1182 chart.series.i[].markPoint.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1183 chart.series.i[].markPoint.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1184 chart.series.i[].markPoint.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1185 chart.series.i[].markPoint.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1186 chart.series.i[].markPoint.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1187 chart.series.i[].markPoint.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.19.1188 chart.series.i[].markPoint.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1189 chart.series.i[].markPoint.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1190 chart.series.i[].markPoint.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.1191 chart.series.i[].markPoint.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1192 chart.series.i[].markPoint.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1193 chart.series.i[].markPoint.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.1194 chart.series.i[].markPoint.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.19.1195 chart.series.i[].markPoint.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 5.19.1196 chart.series.i[].markPoint.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 5.19.1197 chart.series.i[].markPoint.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 5.19.1198 chart.series.i[].markPoint.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1199 chart.series.i[].markPoint.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1200 chart.series.i[].markPoint.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 5.19.1201 chart.series.i[].markPoint.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1202 chart.series.i[].markPoint.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.1203 chart.series.i[].markPoint.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.



## Erlaubte Werte

- normal
- italic
- oblique

## 5.19.1204 chart.series.i[].markPoint.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.1205 chart.series.i[].markPoint.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

## Schlagworte / Tags

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 5.19.1206 chart.series.i[].markPoint.label.emphasis.height

### Datentyp

UNKNOWN

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1207 chart.series.i[].markPoint.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.1208 chart.series.i[].markPoint.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des hervorgehobenen Markierungspunktes.

### 5.19.1209 chart.series.i[].markPoint.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.1210 chart.series.i[].markPoint.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.1211 chart.series.i[].markPoint.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.1212 chart.series.i[].markPoint.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 5.19.1213 chart.series.i[].markPoint.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.1214 chart.series.i[].markPoint.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.1215 chart.series.i[].markPoint.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1216 chart.series.i[].markPoint.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1217 chart.series.i[].markPoint.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1218 chart.series.i[].markPoint.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1219 chart.series.i[].markPoint.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1220 chart.series.i[].markPoint.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1221 chart.series.i[].markPoint.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1222 chart.series.i[].markPoint.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1223 chart.series.i[].markPoint.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1224 chart.series.i[].markPoint.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.1225 chart.series.i[].markPoint.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1226 chart.series.i[].markPoint.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1227 chart.series.i[].markPoint.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 5.19.1228 chart.series.i[].markPoint.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 5.19.1229 chart.series.i[].markPoint.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.1230 chart.series.i[].markPoint.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 5.19.1231 chart.series.i[].markPoint.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1232 chart.series.i[].markPoint.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.1233 chart.series.i[].markPoint.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Markierungspunktes.

### 5.19.1234 chart.series.i[].markPoint.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.1235 chart.series.i[].markPoint.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom

- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.1236 chart.series.i[].markPoint.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.1237 chart.series.i[].markPoint.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.19.1238 chart.series.i[].markPoint.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.19.1239 chart.series.i[].markPoint.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.19.1240 chart.series.i[].markPoint.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1241 chart.series.i[].markPoint.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1242 chart.series.i[].markPoint.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1243 chart.series.i[].markPoint.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1244 chart.series.i[].markPoint.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1245 chart.series.i[].markPoint.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1246 chart.series.i[].markPoint.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1247 chart.series.i[].markPoint.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1248 chart.series.i[].markPoint.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.19.1249 chart.series.i[].markPoint.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 5.19.1250 chart.series.i[].markPoint.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1251 chart.series.i[].markPoint.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Markierungspunkt in einem Diagramm.

### 5.19.1252 chart.series.i[].markPoint.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

### 5.19.1253 chart.series.i[].markPoint.symbol

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'pin' ]

Symbol des Markierungspunktes.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

#### 5.19.1254 chart.series.i[].markPoint.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 5.19.1255 chart.series.i[].markPoint.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

#### Erlaubte Werte

- {integer}
- {percent}

### 5.19.1256 chart.series.i[].markPoint.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 5.19.1257 chart.series.i[].markPoint.symbolSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Symbolgröße. Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

Müssen die Größen der Symbole unterschiedlich sein, kann eine Callback-Funktion im folgenden Format verwendet werden:

```
(value: Array|number, params: Object) => number|Array
```

Der erste Parameter value ist der Wert in data, der zweite Parameter params enthält die restlichen Parameter vom data-Objekt.

### 5.19.1258 chart.series.i[].max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Der spezifiziert maximale Wert.

### 5.19.1259 chart.series.i[].maxOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Maximale Undurchsichtigkeit. Ist gültig für coordinateSystem vom Typ 'geo'.

### 5.19.1260 chart.series.i[].maxSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die gemappte Breite vom maximalen Datenwert max.

Der Wert kann ein absoluter Pixelwert oder ein Prozentwert relativ zur Breite des Layouts sein.

### 5.19.1261 chart.series.i[].min

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der spezifizierte Minimalwert.

### 5.19.1262 chart.series.i[].minAngle

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Minimale Winkel eines Sektors (0 ~ 360). Dieser Wert verhindert, dass Sektoren mit zu kleinen Werten so klein werden, dass die Nutzerinteraktion beeinträchtigt wird.

### 5.19.1263 chart.series.i[].minOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Minimale Undurchsichtigkeit. Ist gültig für coordinateSystem vom Typ 'geo'.

### 5.19.1264 chart.series.i[].minSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die gemappte Breite vom maximalen Datenwert max.

Der Wert kann ein absoluter Pixelwert oder ein Prozentwert relativ zur Breite des Layouts sein. Wenn der Graph des kleinsten Werts kein Dreieck sein soll, kann dieser Wert größer als 0 gesetzt werden.

### 5.19.1265 chart.series.i[].nameMap

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mapping der Namen für angepasste Gebiete. Zum Beispiel:

```
{
  'China' : '??'
}
```

### 5.19.1266 chart.series.i[].nodeClick

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: treemap: 'zoomToNode', sunburst: 'rootToNode' ]

Das Verhalten wenn auf einen Knoten geklickt wird. Mögliche Werte sind:

- false: Nichts passiert.
- In treemap-Graphen: 'zoomToNode': Auf angeklickten Knoten zoomen.
- In sunburst-Graphen: 'rootToNode': Angeklickten Sektor als Wurzel benutzen
- 'link': Wenn link gesetzt ist, leitet die Seite dahin weiter.

## 5.19.1267 chart.series.i[].nodeGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Zwischenraum zwischen allen zwei Rechtecken in jeder Spalte des Graphs.

## 5.19.1268 chart.series.i[].nodes

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Alias von data.

## 5.19.1269 chart.series.i[].nodeScaleRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.6 ]



Zoomrate der Knoten wenn die Maus rein oder raus zoomt. Wenn der Wert auf 0 steht, zoomt der Knoten nicht, wenn die Maus zoomt, verändert sich also in seiner Größe nicht.

### 5.19.1270 chart.series.i[].nodeWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Die Breite der rechteckigen Knoten im Graph.

### 5.19.1271 chart.series.i[].orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Richtung des Layouts im Diagramm.

Mögliche Werte:

Tree (wenn layout 'orthogonal' ist):

- 'LR', 'horizontal': Von links nach rechts.
- 'RL': Von rechts nach links.
- 'TB', 'vertical': Von oben nach unten.
- 'BT': Von unten nach oben.

Sankey:

- 'horizontal': Von links nach rechts.
- 'vertical': Von oben nach unten.

## Erlaubte Werte

- LR
- RL
- TB
- BT
- horizontal
- vertical

### 5.19.1272 chart.series.i[].parallelIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Parallelkoordinaten mit denen kombiniert werden soll, was nützlich für mehrere Parallelachsen in einem Diagramm ist.

### 5.19.1273 chart.series.i[].pointer.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '80%' ]

Die Länge des Zeigers, kann ein absoluter Wert oder ein Prozentwert relativ zum Radius sein.

## Erlaubte Werte

- {integer}
- {percent}

### 5.19.1274 chart.series.i[].pointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeiger des Messdiagramms.

### 5.19.1275 chart.series.i[].pointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Zeiger angezeigt werden soll.

### 5.19.1276 chart.series.i[].pointer.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Die Breite des Zeigers.

### 5.19.1277 chart.series.i[].polarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index des Polarkoordinatensystems mit dem kombiniert werden soll, was nützlich für mehrere Polarachsen in einem Diagramm ist.

### 5.19.1278 chart.series.i[].polyline

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linien als Polygonzüge gezeichnet werden sollen.

Standardmäßig false, wodurch nur eine Linie zwischen zwei Punkten gemalt werden kann.

Wird der Wert auf true gesetzt, akzeptiert data.coords mehr als zwei koordinaten, wodurch ein Polygonzug gezeichnet werden kann. Nützlich wenn Daten aus GPS-Verfolgungen visualisiert werden sollen.

### 5.19.1279 chart.series.i[].progressive

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

progressive spezifiziert die Anzahl der grafischen Elemente die in einem Frame (etwa 16ms) gerendert werden können, wenn progressives Rendering aktiviert ist.

Wenn die Anzahl der Datenpunkte von Tausend bis über 10 Millionen steigt, dauert es zu lange, alle grafischen Elemente zu rendern. Seit ECharts 4 wird progressives Rendering im Workflow unterstützt, wodurch die Daten Block für Block mit jedem Frame verarbeitet und gerendert werden. Somit wird der UI-Thread des Browsers nicht blockiert.

### 5.19.1280 chart.series.i[].progressiveChunkMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mod' ]

Ansatz zum aufteilen der Daten in Blöcke. Mögliche Werte:

- 'sequential': Teile Daten nach Datenindex auf.
- 'mod': Teile Daten nach Modulo auf, wodurch die Daten jedes Blocks aus verschiedenen Bereichen der Datenliste kommen, was einen ansprechenderen visuellen Effekt beim progressiven Rendern geben kann.

#### Erlaubte Werte

- sequential
- mod

### 5.19.1281 chart.series.i[].progressiveThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3000 ]

Wenn die momentane Anzahl DATen über dieser Grenze liegt, wird das progressive Rendering aktiviert.

### 5.19.1282 chart.series.i[].radarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Index der Radarkomponente die das Radadiagramm benutzt.

### 5.19.1283 chart.series.i[].radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Radius des Diagramms.

#### Sunburst-, Kuchendiagramm:

[ Default-Wert: [0, '75%'] ]

Mögliche Werte sind:

- Zahl: Spezifiziert den äußeren Radius direkt.
- String: Zum Beispiel bedeutet '20%', dass der äußere Radius 20% der Größe des Viewports beträgt (des Kleinen zwischen der Breite und Höhe des Diagrammcontainers).

- `Array.<Zahl|String>`: Das erste Element spezifiziert den inneren Radius, das zweite Element spezifiziert den äußeren Radius. Beide folgen den obigen Definitionen.

#### Messdiagramm:

[ Default-Wert: '75%' ]

Kann ein absoluter Wert oder ein Prozentwert relativ zum Kleineren der halben Breite und der halben Höhe des Containers sein.

#### Erlaubte Werte

- {integer}
- {percent}

### 5.19.1284 `chart.series.i[].realtime`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert werden soll.

### 5.19.1285 `chart.series.i[].renderItem.object`

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Nutzerdefinierte Serien verlangen von Entwicklern, ihre eigene Renderlogik zu schreiben. Diese Renderlogik wird als `renderItem` bezeichnet.

Zum Beispiel:

```

var option = {
  ...,
  series: [{
    type: 'custom',
    renderItem: function (params, api) {
      var categoryIndex = api.value(0);
      var start = api.coord([api.value(1), categoryIndex]);
      var end = api.coord([api.value(2), categoryIndex]);
      var height = api.size([0, 1])[1] * 0.6;

      var rectShape = echarts.graphic.clipRectByRect({
        x: start[0],
        y: start[1] - height / 2,
        width: end[0] - start[0],
        height: height
      }, {
        x: params.coordSys.x,
        y: params.coordSys.y,
        width: params.coordSys.width,
        height: params.coordSys.height
      });

      return rectShape && {
        type: 'rect',
        shape: rectShape,
        style: api.style()
      };
    },
    data: data
  ]
}

```

renderItem wird für jedes Datenelement aufgerufen.

renderItem stellt zwei Parameter zur Verfügung:

- params: Gibt Informationen über momentane Serie, Daten und Koordinatensystem.
- api: Enthält einige Methoden.

renderItem sollte die Definition eines grafischen Elements zurückgeben. Siehe renderItem.return

Im Allgemeinen ist der Hauptprozess von renderItem Werte aus data zu holen und sie in grafische Elemente des momentanen Koordinatensystems umzuwandeln. Zwei Methoden in renderItem.arguments.api werden dabei immer benutzt:

- api.value(...) wird verwendet um Werte aus data zu holen. Zum Beispiel gibt api.value(0) den Wert der ersten Dimension des momentanen Datenelements zurück.
- api.coord(...) wird verwendet um Daten in Koordinaten umzuwandeln. Zum Beispiel wandelt var point = api.coord([api.value(0), api.value(1)]) die Daten zu einem Punkt im momentanen Koordinatensystem um.

Manchmal wird api.size(...) benötigt, was die Größe des Koordinatensystems nach einer gegebenen Datenspanne berechnet.

Außerdem kann api.style(...) verwendet werden um Stile zu setzen. Die Methode stellt nicht nur die Einstellungen zur Verfügung, die in series.itemStyle spezifiziert sind, sondern auch das Ergebnis des visuellen Mappings. Diese



Methode kann auch so aufgerufen werden: `api.style({fill: 'green', stroke: 'yellow'})` um diese Stileinstellungen zu überschreiben.

### 5.19.1286 `chart.series.i[].renderLabelForZeroData`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Beschriftungen für leere Daten gerendert werden müssen.

### 5.19.1287 `chart.series.i[].right`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der rechten Seite des Containers

Der Wert von `right` kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Erlaubte Werte

- {integer}
- {percent}

### 5.19.1288 `chart.series.i[].roam`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob das Zoomen und Bewegen mit der Maus aktiviert werden soll, standardmäßig false. Wenn entweder zoomen oder bewegen gewünscht ist, kann der Wert auf 'scale' oder 'move' gesetzt werden. Andernfalls kann er auf true gesetzt werden, um beides zu aktivieren.

#### Erlaubte Werte

- {boolean}
- scale
- move

### 5.19.1289 chart.series.i[].roseType

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Diagramm als Nightingale-Diagramm dargestellt werden soll, welches Daten durch den Radius unterscheidet. Es gibt zwei optionale Werte:

- 'radius': Benutzt den zentralen Winkel um den Prozentanteil der Daten zu zeigen, und den Radius um die Datengröße zu zeigen.
- 'area': Alle Sektoren haben den gleichen zentralen Winkel, Datengröße wird durch den Radius angezeigt.

#### Erlaubte Werte

- {boolean}
- radius
- area

### 5.19.1290 chart.series.i[].sampling

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Downsampling-Strategie, die verwendet wird, wenn die Datengröße deutlich höher als die Pixelgröße ist. Erhöht wenn eingeschaltet die Performanz. Standardmäßig abgeschaltet, wodurch alle Datenpunkte gezeichnet werden.

Mögliche Werte:

- 'average': Nutze durchschnittlichen Wert der Filterpunkte.
- 'max': Nutze maximalen Wert der Filterpunkte.
- 'min': Nutze minimalen Wert der Filterpunkte.
- 'sum': Nutze die Summe der Filterpunkte.

Erlaubte Werte

- average
- max
- min
- sum

### 5.19.1291 chart.series.i[].selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

selectedMode bestimmt, ob multiple Auswahl unterstützt wird. Standardmäßig wird false verwendet, um die Auswahl zu deaktivieren.

Der Wert kann auch 'single', für die Auswahl einzelner Bereiche, oder 'multiple', für die Auswahl mehrer Bereiche sein.

Erlaubte Werte

- {boolean}

- single
- multiple

### 5.19.1292 chart.series.i[].selectedOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Offset-Distanz des ausgewählten Vektors.

### 5.19.1293 chart.series.i[].seriesLayoutBy

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'column' ]

Wenn ein Datenset benutzt wird, bestimmt `seriesLayoutBy`, ob die Spalten oder Zeilen des Datensets auf die Serien gemappt werden. Also, ob die Serien ein Zeilen- oder Spaltenlayout haben. Mögliche Werte:

- 'column': Standardmäßig werden die Spalten des Datensets auf die Serien gemappt. In diesem Fall steht jede Spalte für eine Dimension.
- 'row': Die Zeilen des Datensets werden auf die Serien gemappt. In diesem Fall steht jede Zeile für eine Dimension.

#### Erlaubte Werte

- column
- row

### 5.19.1294 chart.series.i[].showAllSymbol

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Funktioniert nur, wenn die Hauptachse eine Kategorieachse ist (axis.type ist 'category').

Mögliche Werte:

- 'auto': Standardwert. Zeige alle Symbole, wenn genug Platz ist. Folge sonst der Intervallstrategie mit axisLabel.interval.
- true: Zeige alle Symbole.
- false: Folge der Intervallstrategie mit axisLabel.interval.

Erlaubte Werte

- {boolean}
- auto

### 5.19.1295 chart.series.i[].showEffectOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'render' ]

Spezifiziert, wann der Effekt angezeigt werden soll.

**Mögliche Werte:**

- 'render': Zeige den Effekt, wenn das Rendern abgeschlossen ist.
- 'emphasis': Zeige den Effekt, wenn das Element (durch die Maus) hervorgehoben wird.

## Erlaubte Werte

- render
- emphasis

## 5.19.1296 chart.series.i[].showSymbol

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Symbol angezeigt werden soll. Es würde während dem Zeigen auf den Tooltip angezeigt werden.

## 5.19.1297 chart.series.i[].silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

## 5.19.1298 chart.series.i[].silent.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Aktiviere einen Hyperlink-Sprung, wenn auf den Knoten geklickt wird. Dieser Wert ist verfügbar, wenn `series-treemap.nodeClick` auf 'link' steht.

Siehe `series-treemap.data.target`.

### 5.19.1299 `chart.series.i[].silent.object`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

### 5.19.1300 `chart.series.i[].silent.target`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Die gleiche Bedeutung wie `target` im HTML-Label `<a>`, siehe `series-treemap.data.link`. Mögliche Werte sind 'blank' und 'self'.

## Erlaubte Werte

- blank
- self

### 5.19.1301 chart.series.i[].silent.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.19.1302 chart.series.i[].silent.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.19.1303 chart.series.i[].silent.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.19.1304 chart.series.i[].silent.tooltip.extraCSSText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 5.19.1305 chart.series.i[].silent.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

## 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

## 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.19.1306 chart.series.i[].silent.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen der Daten dieser Serie.

### 5.19.1307 chart.series.i[].silent.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.19.1308 chart.series.i[.].silent.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

### 5.19.1309 chart.series.i[.].silent.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.19.1310 chart.series.i[].silent.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1311 chart.series.i[].silent.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.19.1312 chart.series.i[].silent.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.1313 chart.series.i[].silent.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal



- bold
- bolder
- lighter
- {integer}

### 5.19.1314 chart.series.i[.].silent.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1315 chart.series.i[.].silent.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.19.1316 chart.series.i[.].silent.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.data.tooltip funktioniert nur wenn tooltip.trigger auf 'item' steht.

Der Textil des schwebenden Tooltip-Layers.

### 5.19.1317 chart.series.i[.].silent.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.1318 chart.series.i[.].silent.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1319 chart.series.i[.].silent.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1320 chart.series.i[.].silent.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1321 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1322 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1323 chart.series.i[].silent.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1324 chart.series.i[].singleAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index der einzelnen Zeitachse. Standardmäßig 0, da er nur eine Achse enthält.

### 5.19.1325 chart.series.i[].smooth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Kurven geglättet werden sollen.

Wird ein Boolean-Wert angegeben, so bestimmt er, ob Glättung angewendet wird. Wird ein Dezimalwert von 0 bis 1 angegeben, steht er für den Glättungsgrad. Kleinere Werte bedeuten geringere Glättung.

Siehe smoothMonotone um den Glättungsalgorithmus zu verändern.

### 5.19.1326 chart.series.i[].smoothMonotone

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die gebrochene Linie die Monotonität bei der Glättung behält. Der Wert kann auf 'x' oder 'y' gesetzt werden, um die Monotonität auf der X- oder Y-Achse zu behalten. Alternativ kann der Wert auf 'none' gesetzt werden, um einen nicht-monotonen Glättungsalgorithmus zu verwenden.

Der Standardmäßige Glättungsalgorithmus wurde von ECharts 4.0.3 verbessert. Der alte Algorithmus kann verwendet werden, indem smoothMonotone auf 'none' gesetzt wird.

Der alte Algorithmus verwendet den vorherigen und den nächsten Punkt um die Richtung des Kontrollpunkts zu bestimmen, während die Richtung beim neuen Algorithmus immer entweder horizontal (wenn die erste Dimension der Daten monoton ist) oder vertikal (wenn die zweite Dimension der Daten monoton ist) ist.

Der neue Algorithmus funktioniert jedoch nicht mit nicht-monotonen Daten.

Es wird also empfohlen in den meisten Situationen den Standardwert von smoothMonotone zu verwenden. Sind die Daten auf der Y-Achse monoton, sollte der Wert auf 'y' gesetzt werden. Sind die Daten nicht-monoton, sollte der Wert auf 'none' gesetzt werden, um den alten Algorithmus zu verwenden.

#### Erlaubte Werte

- x
- y
- none

### 5.19.1327 chart.series.i[].sort

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Sortiermethode.

**Sunburst:**

[ Default-Wert: 'desc' ]

Sortiermethode die von den Sektoren basierend auf value verwendet wird, was die Summe der Kinder ist, wenn nicht gesetzt. Der Standardwert 'desc' steht für absteigende Reihenfolge. Auch möglich ist 'asc' für aufsteigende Reihenfolge, null für keine Sortierung, oder eine Callback-Funktion wie:

```
function(nodeA, nodeB) {
  return nodeA.getValue() - nodeB.getValue();
}
```

**Funnel:**

[ Default-Wert: 'descending' ]

Datensortierung, akzeptiert die Werte 'ascending', 'descending', 'none' (keine Sortierung), oder eine Funktion nach `Array.prototype.sort(function(a, b) { ... });`

## 5.19.1328 chart.series.i[].splitLine.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Die Länge der Trennlinie. Kann ein Prozentwert relativ zum Radius sein.

## Erlaubte Werte

- {integer}
- {percent}

## 5.19.1329 chart.series.i[].splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #eee ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```



### 5.19.1330 chart.series.i[].splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 5.19.1331 chart.series.i[].splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.19.1332 chart.series.i[].splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.19.1333 chart.series.i[].splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.19.1334 chart.series.i[].splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.19.1335 chart.series.i[].splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.19.1336 chart.series.i[].splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.19.1337 chart.series.i[].splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Breite der Trennlinie.

## 5.19.1338 chart.series.i[].splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Trennlinie

## 5.19.1339 chart.series.i[].splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinie angezeigt werden soll.

## 5.19.1340 chart.series.i[].splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Anzahl der getrennten Segmente der Skala des Messdiagramms.

### 5.19.1341 chart.series.i[].squareRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der erwartete quadratische Schnitt. Das Layout versucht, sich so nah wie möglich an diesen Schnitt anzupassen. Standardmäßig verwendet wird der goldenene Schnitt:  $0.5 * (1 + \text{Math.sqrt}(5))$ .

### 5.19.1342 chart.series.i[].stack

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Name des Stacks. Auf der gleichen Kategorieachse werden Serien mit dem gleichen Stack-Namen aufeinandergelegt.

### 5.19.1343 chart.series.i[].startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Anfangswinkel, dessen Wert zwischen 0 und 360 liegt.

Die Stelle direkt rechts vom Mittelpunkt ist 0 Grad, über dem Mittelpunkt 90 Grad, links vom Mittelpunkt 180 Grad.

### 5.19.1344 chart.series.i[].step

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linie als Schrittlinie dargestellt werden soll. Der Wert 'start', 'middle' oder 'end' sein, was den Schrittpunkt der Schrittlinie bestimmt. Alternativ kann der Wert true oder false sein, wobei true identisch zu 'start' ist.

#### Erlaubte Werte

- {boolean}
- start
- middle
- end

### 5.19.1345 chart.series.i[].stillShowZeroSum

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Sektoren angezeigt werden sollen, wenn alle Daten 0 sind.

### 5.19.1346 chart.series.i[].symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Serie.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

### 5.19.1347 chart.series.i[].symbolBoundingData

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert eine Grenzfläche, die für die grafischen Elemente zur Verfügung steht. Diese Einstellung gibt einen Datenpunkt, welcher dann in eine Koordinate im Koordinatensystem übersetzt wird. Die Koordinate spezifiziert die Grenze. Insbesondere hängt, wenn `symbolBoundingData` gesetzt wurde, die finale Größe (oder das Layout) des grafischen Elements von `symbolBoundingData` ab.

Ist die Referenzleiste horizontal, korrespondiert `symbolBoundingData` zur X-Achse. Ist die Referenzleiste vertikal, korrespondiert `symbolBoundingData` zur Y-Achse.

Regel:

- Wenn `symbolRepeat` nicht verwendet wird:  
`symbolBoundingData` ist standardmäßig so groß wie die Referenzleiste. Die Größe des grafischen Elements wird durch `symbolBoundingData` bestimmt. Zum Beispiel, wenn die Referenzleiste vertikal und ihr Wert 24 ist, `symbolSize` auf `[30, '50%']` und `symbolBoundingData` auf 124 gesetzt ist, wird die endgültige Größe des grafischen Elements  $124 * 50\% = 62$  sein. Ist `symbolBoundingData` nicht gesetzt, wird die endgültige Größe  $24 * 50\% = 12$  sein.
- Wenn `symbolRepeat` verwendet wird:  
`symbolBoundingData` ist der Extremwert des Koordinatensystems. `symbolBoundingData` definiert einen begrenzten Bereich, in dem wiederholte grafische Elemente anhand von `symbolMargin`, `symbolRepeat` und `symbolSize` ausgerichtet werden. Beide dieser Einstellungen bestimmen die Größe der Lücken der wiederholten grafischen Elemente.

`symbolBoundingData` wird normalerweise in den folgenden Fällen verwendet:

- Wenn `symbolClip` verwendet wird:  
Und eine Serie wird verwendet um "Anzahl Wert" anzuzeigen, während eine andere Serie verwendet wird, um "momentaner Wert" anzuzeigen. `symbolBoundingData` kann verwendet werden, um sicherzustellen, dass die grafischen Elemente dieser beiden Serien die gleiche Größe haben.
- Wenn `symbolRepeat` verwendet wird:  
`symbolBoundingData` kann verwendet werden, um sicherzustellen, dass die Lücken zwischen den Elementen in verschiedenen Säulen gleich groß sind. Natürlich kann `symbolBoundingData` auch nicht gesetzt werden, da der Standardwert ein stabiler Wert ist (Extremwert des Koordinatensystems).

`symbolBoundingData` kann auch ein Array sein, wie `[-40, 60]`, welches negative und positive Werte für `symbolBoundingData` spezifiziert.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolBoundingData: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolBoundingData: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolBoundingData: ... // Betrifft nur dieses Datenelement.
  }]
}]
```



### 5.19.1348 chart.series.i[].symbolClip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob grafische Elemente zurechtgeschnitten werden sollen.

- false/null/undefined: Das gesamte grafische Element repräsentiert die Größe von value.
- true: Das zurechtgeschnittene grafische Element repräsentiert die Größe von value.

symbolClip wird normalerweise in diesem Fall benutzt: Sowohl ein maximaler Wert als auch ein momentaner Wert sollen angezeigt werden. In diesem Fall können zwei Serien verwendet werden. Eine für den maximalen Wert im Hintergrund, welche komplette Grafiken verwendet, eine für den momentanen Wert im Vordergrund, welche zurechtgeschnittene Grafiken verwendet.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolClip: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolClip: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolClip: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### 5.19.1349 chart.series.i[].symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 5.19.1350 chart.series.i[].symbolMargin

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den Rand auf beiden Seiten eines grafischen Elements. ("Beide Seiten" bezeichnet die Seiten in Richtung der Wertachse.) Funktioniert nur, wenn symbolRepeat verwendet wird.

Es können absolute Werte (wie 20) verwendet werden, oder Prozentwerte (wie '-30%') relativ zu symbolSize.

symbolMargin kann ein positiver oder negativer Wert sein, was das Überlappen von grafischen Elementen ermöglicht, wenn symbolRepeat verwendet wird.

Ein "!" kann an den Wert angehängt werden, wie '30%!' oder '25!', wodurch eine zusätzliche Lücke an beiden Enden hinzugefügt wird. Sonst erreichen die grafischen Elemente auf beiden Seiten standardmäßig die Grenze.

Hinweis:

- Wenn symbolRepeat auf true/'fixed' gesetzt ist: Der gegebene symbolMargin ist nur ein Referenzwert. Der finale Rand der grafischen Elemente wird anhand von symbolRepeat, symbolMargin und symbolBoundingData berechnet.
- Wenn symbolRepeat auf eine Zahl gesetzt ist: symbolMargin funktioniert nicht mehr.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```

series: [{
  symbolMargin: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolMargin: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolMargin: ... // Betrifft nur dieses Datenelement.
  }]
}]

```

### 5.19.1351 chart.series.i[].symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

#### Erlaubte Werte

- {integer}
- {percent}

### 5.19.1352 chart.series.i[].symbolPatternSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

Ein Bild kann als Muster von grafischen Elementen verwendet werden.

```
var textureImg = new Image();
textureImg.src = 'data:image/jpeg;base64,...'; // dataURI
// Oder
// textureImg.src = 'http://xxx.xxx.xxx/xx.png'; // URL
...
itemStyle: {
  color: {
    image: textureImg,
    repeat: 'repeat'
  }
}
```

symbolPatternSize spezifiziert die Größe des Bildmusters. Ist symbolPatternSize zum Beispiel 400, wird das Bildmuster mit einer Größe von 400px \* 400px dargestellt.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolPatternSize: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolPatternSize: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolPatternSize: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### 5.19.1353 chart.series.i[].symbolRepeat

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob grafische Elemente wiederholt werden sollen. Mögliche Werte:

- false/null/unidentified: Keine Wiederholung. Das heißt, jedes grafische Element repräsentiert je ein Datenelement.
- true: Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen wird anhand der Daten berechnet.
- Eine Zahl: Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen ist immer die gegebene Zahl.
- 'fixed': Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen wird anhand von `symbolBoundingData` berechnet. Das heißt, die Anzahl der Wiederholungen hat nichts mit den Daten zu tun. Diese Option ist nützlich, wenn grafische Elemente als Hintergrund benutzt werden sollen.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolRepeat: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolRepeat: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolRepeat: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### Erlaubte Werte

- {boolean}
- {integer}
- fixed

### 5.19.1354 chart.series.i[].symbolRepeatDirection

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Wenn `symbolRepeat` verwendet wird, spezifiziert `symbolRepeatDirection` die Reihenfolge, in der die wiederholten grafischen Elemente gerendert werden. Diese Einstellung ist in den folgenden Fällen nützlich:

- Wenn `symbolMargin` auf einen negativen Wert gesetzt ist, überlappen sich wiederholte Elemente. `symbolRepeatDirection` kann verwendet werden, um die Reihenfolge der Überlappungen zu bestimmen.
- Wird `animationDelay` oder `animationDelayUpdate` verwendet, bestimmt `symbolRepeatDirection` die Reihenfolge der Indices.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolRepeatDirection: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolRepeatDirection: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolRepeatDirection: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

#### Erlaubte Werte

- start
- end

### 5.19.1355 chart.series.i[].symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 5.19.1356 chart.series.i[].symbolSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße.

Mögliche Werte:

#### **effectScatter:**

Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

#### **line, scatter, radar, tree, graph:**

Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

Müssen die Größen der Symbole unterschiedlich sein, kann eine Callback-Funktion im folgenden Format verwendet werden:

```
(value: Array|number, params: Object) => number|Array
```

Der erste Parameter value ist der Wert in data, der zweite Parameter params enthält die restlichen Parameter vom data-Objekt.

**lines:**

Symbolgröße an den beiden Enden der Linie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren.

Vorsicht: Es können nicht Breite und Höhe getrennt gesetzt werden wie beim normalen SymbolSize.

**pictorialBar:**

Der Wert kann als Array der Form [Breite, Höhe] gesetzt werden. Zum Beispiel steht [20, 10] für eine Breite von 20 und eine Höhe von 10. Er kann auch als einzelne Zahl gesetzt werden, zum Beispiel 10, was äquivalent zu [10, 10] ist.

Absolute Werte (wie 10) und prozentuale Werte (wie '120%', ['55%', 23]) können verwendet werden.

Werden prozentuale Werte verwendet, so wird die endgültige Größe des grafischen Elements anhand der Referenzleiste ausgerechnet.

Als Beispiel gebe es eine Referenzleiste, die auf der X-Achse basiert (selber also eine vertikale Leiste ist) und sei symbolSize auf ['30%', '50%'] gesetzt. Die endgültige Größe des grafischen Elements ist:

- Breite: <Breite der Referenzleiste> \* 30%.
- Höhe:
  - Wenn symbolRepeat verwendet wird: <Höhe der Referenzleiste> \* 50%.
  - Wenn symbolRepeat nicht verwendet wird: <Breite der Referenzleiste> \* 50%.

Der Fall für die Y-Achse verläuft analog.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolSize: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolSize: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolSize: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

## 5.19.1357 chart.series.i[].title.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.19.1358 chart.series.i[].title.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.19.1359 chart.series.i[].title.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.1360 chart.series.i[].title.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1361 chart.series.i[].title.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 5.19.1362 chart.series.i[].title.fontFamily

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1363 chart.series.i[].title.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Schriftgröße.

### 5.19.1364 chart.series.i[].title.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

## Erlaubte Werte

- normal
- italic
- oblique

## 5.19.1365 chart.series.i[].title.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.1366 chart.series.i[].title.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.19.1367 chart.series.i[].title.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 5.19.1368 chart.series.i[].title.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Titel des Messdiagramms.

### 5.19.1369 chart.series.i[].title.offsetCenter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, '-40%'] ]

Die Offset-Position relativ zum Mittelpunkt des Messdiagramms. Das erste Element des Arrays ist der horizontale Offset, das zweite Element ist der vertikale Offset. Kann in absoluten Werten oder Prozentwerten relativ zum Radius des Messdiagramms angegeben werden.

Erlaubte Werte

- {integer}
- {percent}

### 5.19.1370 chart.series.i[].title.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 5.19.1371 chart.series.i[.].title.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

## 5.19.1372 chart.series.i[].title.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.1373 chart.series.i[].title.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.1374 chart.series.i[].title.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1375 chart.series.i[].title.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1376 chart.series.i[].title.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Titel angezeigt werden soll.

### 5.19.1377 chart.series.i[].title.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.1378 chart.series.i[].title.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1379 chart.series.i[].title.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1380 chart.series.i[].title.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1381 chart.series.i[].title.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1382 chart.series.i[].title.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.19.1383 chart.series.i[].title.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1384 chart.series.i[].tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.19.1385 chart.series.i[].tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.19.1386 chart.series.i[].tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.19.1387 chart.series.i[].tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 5.19.1388 chart.series.i[].tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

#### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

#### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

#### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.19.1389 chart.series.i[].tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen dieser Serie.

### 5.19.1390 chart.series.i[].tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:



```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.19.1391 chart.series.i[.].tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

### 5.19.1392 chart.series.i[].tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.19.1393 chart.series.i[].tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1394 chart.series.i[].tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.19.1395 chart.series.i[].tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.1396 chart.series.i[].tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 5.19.1397 chart.series.i[].tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1398 chart.series.i[].tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.19.1399 chart.series.i[].tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Textstil des schwebenden Tooltip-Layers.

### 5.19.1400 chart.series.i[].tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.19.1401 chart.series.i[].tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1402 chart.series.i[].tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1403 chart.series.i[].tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 5.19.1404 chart.series.i[].tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 5.19.1405 chart.series.i[].tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.19.1406 chart.series.i[].tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1407 chart.series.i[].top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.19.1408 chart.series.i[].upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 5.19.1409 chart.series.i[].upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1410 chart.series.i[].upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1411 chart.series.i[].upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.19.1412 chart.series.i[].upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.19.1413 chart.series.i[].upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1414 chart.series.i[].upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 5.19.1415 chart.series.i[].upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 5.19.1416 chart.series.i[].upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.19.1417 chart.series.i[].upperLabel.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.19.1418 chart.series.i[.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.19.1419 chart.series.i[.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.19.1420 chart.series.i[.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 5.19.1421 chart.series.i[].upperLabel.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Höhe des Bereichs der Beschriftung.

### 5.19.1422 chart.series.i[].upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.19.1423 chart.series.i[.].upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

upperLabel wird verwendet, um zu spezifizieren, ob eine Beschriftung angezeigt werden soll, wenn der Knoten Kinder hat. Wenn upperLabel.show auf true gesetzt ist, wird die Beschriftung von Elternknoten angezeigt.

So wie series-treemap.label kann die Option upperLabel direkt an der Wurzel von series-treemap definiert werden, in series-treemap.level, oder in jedem Element von series-treemap.data.

Anders gesagt spezifiziert series-treemap.label den Stil der Beschriftung von Blattknoten, während series-treemap.upperLabel den Stil der Beschriftung von Knoten mit Kindern spezifiziert. Die Beschriftung wird dann im oberen inneren Bereich des Knotens angezeigt.

Tips: In treemap kann das upperLabel Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.1424 chart.series.i[].upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 5.19.1425 chart.series.i[].upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.19.1426 chart.series.i[].upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

#### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

#### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 5.19.1427 chart.series.i[.].upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.19.1428 chart.series.i[.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 5.19.1429 chart.series.i[].upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 5.19.1430 chart.series.i[].upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 5.19.1431 chart.series.i[].upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.19.1432 chart.series.i[].upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.19.1433 chart.series.i[].upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.19.1434 chart.series.i[].upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.19.1435 chart.series.i[].upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.19.1436 chart.series.i[].upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.19.1437 chart.series.i[].upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.19.1438 chart.series.i[.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.19.1439 chart.series.i[.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.



### 5.19.1440 chart.series.i[].upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.19.1441 chart.series.i[].upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.19.1442 chart.series.i[].visibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Ein Knoten wird nicht angezeigt wenn die Größe seines Bereichs kleiner als dieser Wert ist (Einheit: px quadrat).

So werden kleine Knoten, die ansonsten eng aneinander stehen würden, versteckt. Wenn ein Nutzer die treemap vergrößert, vergrößert sich auch die Bereichsgröße, und das Rechteck wird angezeigt, wenn sie über diese Grenze steigt.

Siehe serie-treemap.levels für Details zu visuellem Encoding.

Tips: In treemap kann das visibleMin Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.1443 chart.series.i[].visualDimension

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

treemap kann beliebige Dimensionen der Daten ins Visuelle mappen.

Der Wert von `series-treemap.data` kann ein Array sein. Jeder Eintrag des Arrays steht für eine "dimension". `visualDimension` spezifiziert die Dimension, auf der visuelles Mapping durchgeführt wird.

Siehe `series-treemap.levels` für Details zu visuellem Encoding.

Tips: In `treemap` kann das `visualDimension` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 5.19.1444 `chart.series.i[].visualMax`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn `colorMappingBy` auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von `visualMin` oder `visualMax` bestimmt werden.

### 5.19.1445 `chart.series.i[].visualMin`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Der minimale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn colorMappingBy auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von visualMin oder visualMax bestimmt werden.

### 5.19.1446 chart.series.i[].width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite der Komponente.

### 5.19.1447 chart.series.i[].xAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der X-Achse mit der kombiniert werden soll, nützlich für mehrere X-Achsen in einem Diagramm.

### 5.19.1448 chart.series.i[].yAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der X-Achse mit der kombiniert werden soll, nützlich für mehrere X-Achsen in einem Diagramm.

### 5.19.1449 chart.series.i[].z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.19.1450 chart.series.i[].zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.19.1451 chart.series.i[.].zoom

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Zoom-Rate des momentanen Viewports.

### 5.19.1452 chart.series.i[.].zoomToNodeRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.32\*0.32 ]

Die treemap wird automatisch auf eine angemessene Rate gezoomt, wenn ein Knoten angeklickt wird (solange nodeClick auf 'zoomToNode' gesetzt ist und kein Drill-Down durchgeführt wird). Diese Einstellung gibt die Rate an.

### 5.19.1453 chart.series.i[.].silent.children

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kindknoten, rekursive definition. Konfigurationen sind identisch zu denen in series.data

### 5.19.1454 chart.series.fixed.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sunburst' ]

Wird in Sunburst-Diagrammen verwendet, um einen Namen für die Serie bereitzustellen.

Example:

```
dashlet.addOption("chart.series.fixed.name", "Gesamt");
```

### 5.19.1455 chart.series.i[].name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	gantt, series-name

Name für eine Series

Beispiel:

```
// For Gantt chart, give name for the different series, to be used in click-function
(for example) dashlet.addOption("chart.series.n[ganttitem].name", "cmganttdiagramm");
dashlet.addOption("chart.series.n[ganttyaxis1].name", "GantBars");
dashlet.addOption("chart.series.n[ganttyaxis2].name", "pdficon");
```

### 5.19.1456 chart.series.i[].label.alignTo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Default: none

Label-Ausrichtungsrichtlinie.

- 'none' (default): Etikettenzeilen haben eine feste Länge als `labelLine.length` und `labelLine.length2`
- 'labelLine': Die Ausrichtung am Ende der Labelzeilen und die Länge der kürzesten horizontalen Labelzeilen wird durch `labelLine.length2` konfiguriert.
- 'edge': Die Ausrichtung auf Text und der Abstand zwischen den Rändern des Textes und dem Ansichtsfenster wird durch `label.edgeDistance` konfiguriert.

#### Erlaubte Werte

- labelLine
- edge
- none

### 5.19.1457 chart.series.i[].label.edgeDistance

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	



Der horizontale Abstand zwischen Textkanten und Ansichtsfenster, wenn label.position „outer“ und label.alignTo „edge“ ist.

In den meisten Fällen benötigen Sie einen kleinen edgeDistance-Wert wie 10 für mobile Geräte, um sicherzustellen, dass mehr Text angezeigt werden kann, anstatt .... Bei größeren Auflösungen sollte jedoch ein Prozentwert angewendet werden, damit die Beschriftungszeilen nicht angezeigt werden zu lang. Wenn Ihr Diagramm in unterschiedlichen Auflösungen verwendet wird, ist es ratsam, responsives Design für unterschiedliche Auflösungen festzulegen.

### 5.19.1458 chart.series.i[].label.bleedMargin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Der horizontale Abstand zwischen Text und Ansichtsfenster, wenn label.position „outer“ und label.alignTo „none“ oder „labelLine“ ist. Längerer Text wird in '...' abgeschnitten.

### 5.19.1459 chart.series.i[].label.distanceToLabelLine

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Abstand zwischen Beschriftungslinie und Text.

### 5.19.1460 chart.series.i[].minShowLabelAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Wenn ein Sektor kleiner als dieser Winkel ist (0 ~ 360), werden Label und LabelLine nicht angezeigt.

## 5.20 ECharts (4) Single Axis

### 5.20.1 chart.singleAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Eine Achse mit einer einzelnen Dimension. Kann verwendet werden um Daten in einer Dimension darzustellen.

### 5.20.2 chart.singleAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.20.3 chart.singleAxis.zlevel

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

#### 5.20.4 chart.singleAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

#### 5.20.5 chart.singleAxis.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- right
- center

### 5.20.6 chart.singleAxis.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle

- bottom

### 5.20.7 chart.singleAxis.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Erlaubte Werte

- {percent}
- {integer}

### 5.20.8 chart.singleAxis.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Erlaubte Werte

- {percent}

- {integer}

### 5.20.9 chart.singleAxis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Grid-Komponente. Standardmäßig adaptiv.

### 5.20.10 chart.singleAxis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Grid-Komponente. Standardmäßig adaptiv.

### 5.20.11 chart.singleAxis.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'horizontal' ]

Orientierung der Achse. Standardmäßig ist sie horizontal, man kann diesen Wert auf 'vertical' setzen um eine vertikale Achse zu erhalten.

#### Erlaubte Werte

- horizontal
- vertical

## 5.20.12 chart.singleAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

#### Erlaubte Werte

- value
- category
- time
- log

## 5.20.13 chart.singleAxis.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

### 5.20.14 chart.singleAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 5.20.15 chart.singleAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.



### 5.20.16 chart.singleAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 5.20.17 chart.singleAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

Erlaubte Werte

- normal
- italic
- oblique

### 5.20.18 chart.singleAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.20.19 chart.singleAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie des Achsennamens.

Akzeptiert auch 'serif' , 'monospace' , ...

### 5.20.20 chart.singleAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 5.20.21 chart.singleAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.20.22 chart.singleAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 5.20.23 chart.singleAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.20.24 chart.singleAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.20.25 chart.singleAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.20.26 chart.singleAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.20.27 chart.singleAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.20.28 chart.singleAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.20.29 chart.singleAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.20.30 chart.singleAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.20.31 chart.singleAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.20.32 chart.singleAxis.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.20.33 chart.singleAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.34 chart.singleAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.35 chart.singleAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.20.36 chart.singleAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.20.37 chart.singleAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.20.38 chart.singleAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.20.39 chart.singleAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.20.40 chart.singleAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.20.41 chart.singleAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

#### 5.20.42 chart.singleAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 5.20.43 chart.singleAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 5.20.44 chart.singleAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert werden soll. Neue Option aus ECharts 3.

### 5.20.45 chart.singleAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

#### 5.20.46 chart.singleAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.20.47 chart.singleAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 5.20.48 chart.singleAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 5.20.49 chart.singleAxis.splitNumber

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 5.20.50 chart.singleAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 5.20.51 chart.singleAxis.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 5.20.52 chart.singleAxis.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 5.20.53 chart.singleAxis.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 5.20.54 chart.singleAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 5.20.55 chart.singleAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 5.20.56 chart.singleAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 5.20.57 chart.singleAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 5.20.58 chart.singleAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 5.20.59 chart.singleAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 5.20.60 chart.singleAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 5.20.61 chart.singleAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.20.62 chart.singleAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 5.20.63 chart.singleAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.20.64 chart.singleAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.65 chart.singleAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.66 chart.singleAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.67 chart.singleAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.68 chart.singleAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.69 chart.singleAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 5.20.70 chart.singleAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 5.20.71 chart.singleAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 5.20.72 chart.singleAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.20.73 chart.singleAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 5.20.74 chart.singleAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 5.20.75 chart.singleAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenticks.

### 5.20.76 chart.singleAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks. Standardmäßig auf axisLine.lineStyle.color gesetzt.

### 5.20.77 chart.singleAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Achsenticks.

### 5.20.78 chart.singleAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Achsenticks.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.20.79 chart.singleAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.80 chart.singleAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.81 chart.singleAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.82 chart.singleAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.83 chart.singleAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.84 chart.singleAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenbeschriftung gehören.

### 5.20.85 chart.singleAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 5.20.86 chart.singleAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 5.20.87 chart.singleAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 5.20.88 chart.singleAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 5.20.89 chart.singleAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 5.20.90 chart.singleAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
Achenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
Achenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.20.91 chart.singleAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.20.92 chart.singleAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 5.20.93 chart.singleAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 5.20.94 chart.singleAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.20.95 chart.singleAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 5.20.96 chart.singleAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 5.20.97 chart.singleAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.20.98 chart.singleAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.20.99 chart.singleAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

#### 5.20.100 chart.singleAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.20.101 chart.singleAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.20.102 chart.singleAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.20.103 chart.singleAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.20.104 chart.singleAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.20.105 chart.singleAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.20.106 chart.singleAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.20.107 chart.singleAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.20.108 chart.singleAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.20.109 chart.singleAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.20.110 chart.singleAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.111 chart.singleAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.112 chart.singleAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.20.113 chart.singleAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.20.114 chart.singleAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.20.115 chart.singleAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.20.116 chart.singleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.20.117 chart.singleAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.20.118 chart.singleAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.20.119 chart.singleAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 5.20.120 chart.singleAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 5.20.121 chart.singleAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.20.122 chart.singleAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinien.

### 5.20.123 chart.singleAxis.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 5.20.124 chart.singleAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinien.

### 5.20.125 chart.singleAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.20.126 chart.singleAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.127 chart.singleAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.128 chart.singleAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.129 chart.singleAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.130 chart.singleAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.131 chart.singleAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 5.20.132 chart.singleAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 5.20.133 chart.singleAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 5.20.134 chart.singleAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 5.20.135 chart.singleAxis.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbararray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 5.20.136 chart.singleAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.137 chart.singleAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.138 chart.singleAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.139 chart.singleAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.140 chart.singleAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.141 chart.singleAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 5.20.142 chart.singleAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 5.20.143 chart.singleAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 5.20.144 chart.singleAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.20.145 chart.singleAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.20.146 chart.singleAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 5.20.147 chart.singleAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird die Beschriftung nicht angezeigt. Ist aber tooltip.axisPointer.type auf 'cross' gesetzt, so wird die Beschriftung automatisch angezeigt.

### 5.20.148 chart.singleAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes in der Beschriftung. Wird standardmäßig automatisch bestimmt. Der Wert kann auch auf '2' gesetzt werden, was anzeigt, dass zwei Stellen hinter dem Komma reserviert werden.

### 5.20.149 chart.singleAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.20.150 chart.singleAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 5.20.151 chart.singleAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 5.20.152 chart.singleAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 5.20.153 chart.singleAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.20.154 chart.singleAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.20.155 chart.singleAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.20.156 chart.singleAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.20.157 chart.singleAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.158 chart.singleAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.159 chart.singleAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.20.160 chart.singleAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.20.161 chart.singleAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.



## 5.20.162 chart.singleAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 5.20.163 chart.singleAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 5.20.164 chart.singleAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.20.165 chart.singleAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.20.166 chart.singleAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.20.167 chart.singleAxis.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.20.168 chart.singleAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.20.169 chart.singleAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.170 chart.singleAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.171 chart.singleAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.172 chart.singleAxis.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.173 chart.singleAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type auf 'line' steht.

### 5.20.174 chart.singleAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 5.20.175 chart.singleAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

## 5.20.176 chart.singleAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.20.177 chart.singleAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.178 chart.singleAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.179 chart.singleAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.180 chart.singleAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.181 chart.singleAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.182 chart.singleAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type auf 'shadow' steht.

### 5.20.183 chart.singleAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.20.184 chart.singleAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.185 chart.singleAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.186 chart.singleAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.187 chart.singleAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.188 chart.singleAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.189 chart.singleAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 5.20.190 chart.singleAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 5.20.191 chart.singleAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 5.20.192 chart.singleAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 5.20.193 chart.singleAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um den Handle zu verwenden.

### 5.20.194 chart.singleAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon des Handles.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.20.195 chart.singleAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 5.20.196 chart.singleAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 5.20.197 chart.singleAxis.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 5.20.198 chart.singleAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 5.20.199 chart.singleAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.200 chart.singleAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.201 chart.singleAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.202 chart.singleAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.203 chart.singleAxis.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Koordinatensystems.

#### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

### 5.20.204 chart.singleAxis.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

### 5.20.205 chart.singleAxis.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentyten in grid und polar unterstützt. Weiterhin lassen sich Achsen über axisPointer.axis zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item
- axis
- none

### 5.20.206 chart.singleAxis.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Konfigurationselement für den Achsenindikator.

`tooltip.axisPointer` ist wie syntaktischer Zucker von `axisPointer`-Einstellungen an Achsen (zum Beispiel `xAxis.axisPointer` oder `angleAxis.axisPointer`). Detailliertere Einstellungen können an `someAxis.axisPointer` vorgenommen werden. Im Normalfall ist die Verwendung von `tooltip.axisPointer` jedoch praktischer.

**Hinweis:** Konfigurationen an `tooltip.axisPointer` haben eine niedrigere Priorität als `someAxis.axisPointer`

## 5.20.207 `chart.singleAxis.tooltip.axisPointer.type`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatorotyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei `axisPointers` auf zwei orthometrischen Achsen zu aktivieren.

Erlaubte Werte

- line
- shadow
- none
- cross

## 5.20.208 `chart.singleAxis.tooltip.axisPointer.axis`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem selbst die Achsen, welche ihren Achsenpointer anzeigen sollen (Standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

#### Erlaubte Werte

- x
- y
- radius
- angle

### 5.20.209 chart.singleAxis.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.20.210 chart.singleAxis.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.20.211 chart.singleAxis.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des axisPointers.

### 5.20.212 chart.singleAxis.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.20.213 chart.singleAxis.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.20.214 chart.singleAxis.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:



```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 5.20.215 chart.singleAxis.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 5.20.216 chart.singleAxis.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.20.217 chart.singleAxis.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 5.20.218 chart.singleAxis.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.20.219 chart.singleAxis.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.20.220 chart.singleAxis.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.20.221 chart.singleAxis.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.20.222 chart.singleAxis.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.223 chart.singleAxis.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.224 chart.singleAxis.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.20.225 chart.singleAxis.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.20.226 chart.singleAxis.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.20.227 chart.singleAxis.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.20.228 chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.20.229 chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.20.230 chart.singleAxis.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.20.231 chart.singleAxis.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.20.232 chart.singleAxis.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.20.233 chart.singleAxis.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.20.234 chart.singleAxis.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.235 chart.singleAxis.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.236 chart.singleAxis.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.237 chart.singleAxis.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.238 chart.singleAxis.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 5.20.239 chart.singleAxis.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

## 5.20.240 chart.singleAxis.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

## 5.20.241 chart.singleAxis.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.20.242 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.243 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.244 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.245 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.246 chart.singleAxis.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.247 chart.singleAxis.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 5.20.248 chart.singleAxis.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.20.249 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.250 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.251 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.252 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.253 chart.singleAxis.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.254 chart.singleAxis.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

## 5.20.255 chart.singleAxis.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.20.256 chart.singleAxis.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.20.257 chart.singleAxis.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.20.258 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.20.259 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.20.260 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.20.261 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.20.262 chart.singleAxis.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.20.263 chart.singleAxis.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:



```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 5.20.264 chart.singleAxis.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datename, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datename, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.20.265 chart.singleAxis.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.20.266 chart.singleAxis.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.20.267 chart.singleAxis.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.20.268 chart.singleAxis.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.20.269 chart.singleAxis.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 5.20.270 chart.singleAxis.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.20.271 chart.singleAxis.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.20.272 chart.singleAxis.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

## 5.20.273 chart.singleAxis.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.20.274 chart.singleAxis.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.20.275 chart.singleAxis.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:



```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.20.276 chart.singleAxis.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.277 chart.singleAxis.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.20.278 chart.singleAxis.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.20.279 chart.singleAxis.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.20.280 chart.singleAxis.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.20.281 chart.singleAxis.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.20.282 chart.singleAxis.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.20.283 chart.singleAxis.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.20.284 chart.singleAxis.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

## 5.21 ECharts (4) Text Style

### 5.21.1 chart.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Globaler Schriftstil.

### 5.21.2 chart.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.21.3 chart.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.21.4 chart.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 5.21.5 chart.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.21.6 chart.textStyle.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.21.7 chart.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.21.8 chart.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.21.9 chart.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.21.10 chart.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.21.11 chart.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.21.12 chart.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.21.13 chart.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.21.14 chart.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.21.15 chart.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 5.22 ECharts (4) Timeline

### 5.22.1 chart.timeline.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeistrahle-Komponente, welche Funktionen bereitstellt wie das Wechseln und das Abspielen von ECharts Optionen.

Anders als andere Komponenten braucht die Zeitstrahl-Komponente mehrere Optionen. Wenn die traditionelle Art der ECharts Option als *atomic option* bezeichnet wird, dann sollte die für Zeitstrahlen verwendete Option *compound option* genannt werden, da sie aus mehreren zusammengestellten atomic options besteht. Zum Beispiel:

```
// Im folgenden Beispiel ist baseOption eine "atomic option", genau wie alle Einträge
// im options Array.
// Jede der atomic options folgt einer Konfiguration, die in diesem Dokument
// eingeführt wird.
myChart.setOption(
  {
    baseOption: {
      timeline: {
        ...,
        data: ['2002-01-01', '2003-01-01', '2004-01-01']
      },
      title: {
        subtext: ' Daten sind vom National Bureau of Statistics '
      },
      grid: {...},
      xAxis: [...],
      yAxis: [...],
      series: [
        { // Andere Einstellungen von Serie 1
          type: 'bar',
          ...
        },
        { // Andere Einstellungen von Serie 2
          type: 'line',
          ...
        },
        { // Andere Einstellungen von Serie 3
          type: 'pie',
          ...
        }
      ]
    },
    options: [
      { // Option, die zu '2002-01-01' gehört
        title: {
          text: 'Die Statistiken aus dem Jahr 2002'
        },
        series: [
          {data: []}, // Die Daten von Serie 1
          {data: []}, // Die Daten von Serie 2
          {data: []} // Die Daten von Serie 3
        ]
      },
      { // Option, die zu '2003-01-01' gehört
        title: {
          text: 'Die Statistiken aus dem Jahr 2003'
        },
        series: [
          {data: []},
          {data: []},
          {data: []}
        ]
      }
    ]
  }
)
```

```

    },
    { // Option, die zu '2004-01-01' gehört
      title: {
        text: 'Die Statistiken aus dem Jahr 2004'
      },
      series: [
        {data: []},
        {data: []},
        {data: []}
      ]
    }
  ]
}
);

```

Im obigen Beispiel gehört jeder Eintrag in `timeline.data` zu einer Option in Array von `options`.

### Hinweise und Best Practices

- Geteilte Einstellungen sollten in `baseOption` gesetzt werden. Wenn der Punkt auf dem Zeitstrahl gewechselt wird, wird die zugehörige Option im `options` Array mit `baseOption` kombiniert, um die Endgültige Option zu erhalten.
- Ist ein Attribut in einer Option im `option` Array konfiguriert, sollte es auch bei den anderen Optionen konfiguriert sein. Ist es das nicht, wird es ignoriert.
- Optionen in *compound options* unterstützen keine Vereinigung. Das bedeutet, dass beim erneuten Aufruf von `chart.setOption(rawOption)`, falls `rawOption` eine *compound option* ist (also ein `options` Array enthält), die neuen `rawOption.options` die alten ersetzen, statt sich mit ihnen zu vereinen. Das neue `rawOption.baseOption` wird hingegen mit dem Alten vereint.

### Kompatibilität mit ECharts 2:

- ECharts 3 unterstützt den Parameter `timeline.notMerge` nicht mehr, und somit implizit auch den *notMerge-Modus* nicht. Wird diese Funktion gebraucht, kann die Option im eigenen Programm bearbeitet werden, bevor sie an `setOption(option, true)` übergeben wird.
- Im Vergleich zwischen ECharts 3 und ECharts 2 ist der Definitionsort der Zeitstrahl-Attribute ein Anderer. In ECharts 3 wurde er zu `baseOption` umgelegt und wird als separate Komponente angesehen, welche auch mit dem Definitionsort von ECharts 2 kompatibel ist, was aber nicht empfohlen wird.

## 5.22.2 chart.timeline.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Zeitstrahl-Komponente angezeigt werden soll. Steht der Wert auf false, wird sie nicht angezeigt, führt aber trotzdem ihre Funktionen aus.

### 5.22.3 chart.timeline.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'slider' ]

Dieses Attribut hat momentan 'slider' als einzigen gültigen Wert. Muss nicht verändert werden.

### 5.22.4 chart.timeline.axisType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'time' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.

Erlaubte Werte

- value
- category

- time

### 5.22.5 chart.timeline.currentIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Zeigt an, welches Element momentan ausgewählt ist. Ist currentIndex zum Beispiel 0, so ist der momentan ausgewählte Eintrag timeline.data[0] (unter Verwendung von options[0]).

### 5.22.6 chart.timeline.autoPlay

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die automatische Abspielung aktiviert ist.

### 5.22.7 chart.timeline.rewind

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob rückwärts Abspulen unterstützt wird.

### 5.22.8 chart.timeline.loop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Abspielen sich wiederholen soll.

### 5.22.9 chart.timeline.playInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Gibt die Abspielgeschwindigkeit (Zeit zwischen den Wechseln) in Millisekunden an.

### 5.22.10 chart.timeline.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert wird, während der Kontrollpunkt sich bewegt.

### 5.22.11 chart.timeline.controlPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'left' ]

Position des "Abspielen"-Knopfes

Erlaubte Werte

- left
- right

### 5.22.12 chart.timeline.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.22.13 chart.timeline.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.22.14 chart.timeline.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- left
- center
- right
- {percent}
- {integer}

### 5.22.15 chart.timeline.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

### 5.22.16 chart.timeline.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 5.22.17 chart.timeline.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 5.22.18 chart.timeline.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.22.19 chart.timeline.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Ausrichtung der Komponente, gültige Werte sind:

- 'vertical': Vertikales Layout.
- 'horizontal': Horizontales Layout.

#### Erlaubte Werte

- vertical
- horizontal

### 5.22.20 chart.timeline.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Zeitstrahl-Komponente umgekehrt werden soll, wodurch das erste Element zum letzten wird usw.

### 5.22.21 chart.timeline.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'emptyCircle' ]

Symbol des Zeitstrahls.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- emptyCircle
- rect
- roundRect
- triangle

- diamond
- pin
- arrow
- none

### 5.22.22 chart.timeline.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe des Zeitstrahl-Symbols. Der Wert kann als einzelne Zahl gesetzt werden, wie 10, oder als Array welches für Breite und Höhe steht. Zum Beispiel: [20, 10] steht für ein Symbol mit Breite 20 und Höhe 10.

### 5.22.23 chart.timeline.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationgrad des Zeitstrahl-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 5.22.24 chart.timeline.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 5.22.25 chart.timeline.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 5.22.26 chart.timeline.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achse des Zeitstrahls.



### 5.22.27 chart.timeline.lineStyle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie angezeigt werden soll. Kann auf false gesetzt werden um die Achsenlinie zu verstecken um einen anderen Stil zu verwenden.

### 5.22.28 chart.timeline.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Linienfarbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.22.29 chart.timeline.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 2 ]

Linienbreite des Zeitstrahls.

## 5.22.30 chart.timeline.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp des Zeitstrahls.

### Erlaubte Werte

- solid
- dashed
- dotted

## 5.22.31 chart.timeline.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.22.32 chart.timeline.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.22.33 chart.timeline.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.22.34 chart.timeline.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.22.35 chart.timeline.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.22.36 chart.timeline.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenbeschriftung, emphasis ist der Stil des hervorgehobenen Textes. Der Textstil in emphasis wird zum Beispiel verwendet, wenn die Maus auf den Text zeigt oder die Legende eine Verbindung schließt.

### 5.22.37 chart.timeline.label.position

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Mögliche Werte:

- 'auto': Automatisches Layout.
- 'left': An den linken Rand stellen. Gültig wenn timeline.orient auf 'horizontal' steht.
- 'right': An den rechten Rand stellen. Gültig wenn timeline.orient auf 'horizontal' steht.
- 'top': An den oberen Rand stellen. Gültig wenn timeline.orient auf 'vertical' steht.
- 'bottom': An den unteren Rand stellen. Gültig wenn timeline.orient auf 'vertical' steht.
- Nummer: Wird ein numerischer Wert zugewiesen, steht dieser für den Freiraum zwischen Beschriftung und Achse. Ist der Wert 0, steht die Beschriftung an der gleichen Stelle wie die Achse. Negative Werte sind gültig, um Freiraum in die andere Richtung zu definieren.

Erlaubte Werte

- auto
- left
- right
- top
- bottom
- {integer}

### 5.22.38 chart.timeline.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.22.39 chart.timeline.label.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Intervall der Beschriftungen. Wird der Wert auf einen numerischen Wert wie 2 gesetzt, wird alle zwei Elemente eine Beschriftung angezeigt.

### 5.22.40 chart.timeline.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.22.41 chart.timeline.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achsenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Verwendung von String-Templates; Die Template-Variablen {value} ist die
// standardmäßige Beschriftung der Achse
formatter: '{value} kg'

// Verwendung von Callback-Funktionen; Funktionsparameter sind Achsenwert und
// Achsenindex
formatter: function (value, index) {
  // Formatiert auf Monat/Tag, zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getFullYear());
  }
  return texts.join('/');
}
```

### 5.22.42 chart.timeline.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#304654" ]

Textfarbe.

### 5.22.43 chart.timeline.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

#### 5.22.44 chart.timeline.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

#### 5.22.45 chart.timeline.label.fontFamily

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

timeline.lable.normal Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.22.46 chart.timeline.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.22.47 chart.timeline.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 5.22.48 chart.timeline.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.22.49 chart.timeline.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.22.50 chart.timeline.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 5.22.51 chart.timeline.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 5.22.52 chart.timeline.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.22.53 chart.timeline.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.22.54 chart.timeline.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.22.55 chart.timeline.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.22.56 chart.timeline.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.22.57 chart.timeline.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.22.58 chart.timeline.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.22.59 chart.timeline.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.22.60 chart.timeline.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.22.61 chart.timeline.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 5.22.62 chart.timeline.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 5.22.63 chart.timeline.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.22.64 chart.timeline.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.22.65 chart.timeline.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.22.66 chart.timeline.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.22.67 chart.timeline.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.22.68 chart.timeline.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen Zeitstrahl-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 5.22.69 chart.timeline.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 5.22.70 chart.timeline.label.emphasis.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Intervall der Beschriftungen. Wird der Wert auf einen numerischen Wert wie 2 gesetzt, wird alle zwei Elemente eine Beschriftung angezeigt.

### 5.22.71 chart.timeline.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 5.22.72 chart.timeline.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achsenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Verwendung von String-Templates; Die Template-Variable {value} ist die
// standardmäßige Beschriftung der Achse
formatter: '{value} kg'

// Verwendung von Callback-Funktionen; Funktionsparameter sind Achsenwert und
// Achsenindex
formatter: function (value, index) {
  // Formatiert auf Monat/Tag, zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 5.22.73 chart.timeline.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Textfarbe.

### 5.22.74 chart.timeline.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.22.75 chart.timeline.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.22.76 chart.timeline.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

timeline.lable.emphasis Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.22.77 chart.timeline.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.



### 5.22.78 chart.timeline.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.22.79 chart.timeline.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 5.22.80 chart.timeline.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.22.81 chart.timeline.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.22.82 chart.timeline.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 5.22.83 chart.timeline.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 5.22.84 chart.timeline.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 5.22.85 chart.timeline.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.22.86 chart.timeline.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.22.87 chart.timeline.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.22.88 chart.timeline.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.22.89 chart.timeline.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.22.90 chart.timeline.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.22.91 chart.timeline.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.22.92 chart.timeline.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.22.93 chart.timeline.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.22.94 chart.timeline.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.22.95 chart.timeline.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.



### 5.22.96 chart.timeline.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.22.97 chart.timeline.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.22.98 chart.timeline.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 5.22.99 chart.timeline.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grafikstil des Zeitstrahls, emphasis ist der Stil des hervorgehobenen Elements, der zum Beispiel verwendet wird, wenn die Maus auf das Objekt zeigt oder die Legende eine Verbindung schließt.

## 5.22.100 chart.timeline.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Farbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.22.101 chart.timeline.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 5.22.102 chart.timeline.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 5.22.103 chart.timeline.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 5.22.104 chart.timeline.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.22.105 chart.timeline.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.22.106 chart.timeline.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.22.107 chart.timeline.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.22.108 chart.timeline.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.22.109 chart.timeline.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil von Element des Zeitstrahls im hervorgehobenen Zustand.

### 5.22.110 chart.timeline.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Mit Textur füllen
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.22.111 chart.timeline.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.22.112 chart.timeline.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 5.22.113 chart.timeline.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 5.22.114 chart.timeline.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.22.115 chart.timeline.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.22.116 chart.timeline.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.22.117 chart.timeline.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.22.118 chart.timeline.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.22.119 chart.timeline.checkpointStyle.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des ausgewählten Zeitstrahl-Elements (Checkpoint).

### 5.22.120 chart.timeline.checkpointStyle.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'circle' ]

Symbol des Checkpoints.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- emptyCircle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

### 5.22.121 chart.timeline.checkpointStyle.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe des Checkpoint-Symbols. Der Wert kann als einzelne Zahl gesetzt werden, wie 10, oder als Array welches für Breite und Höhe steht. Zum Beispiel: [20, 10] steht für ein Symbol mit Breite 20 und Höhe 10.

### 5.22.122 chart.timeline.checkpointStyle.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 5.22.123 chart.timeline.checkpointStyle.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 5.22.124 chart.timeline.checkpointStyle.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 5.22.125 chart.timeline.checkpointStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Checkpoints in der Zeitstrahl-Komponente.

### 5.22.126 chart.timeline.checkpointStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Rahmenbreite des Checkpoints in der Zeitstrahl-Komponente.

### 5.22.127 chart.timeline.checkpointStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(194, 53, 49, 0.5)' ]

Die Rahmenfarbe des Checkpoints in der Zeitstrahl-Komponente.

### 5.22.128 chart.timeline.checkpointStyle.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob in der Zeitstrahl-Komponente die Bewegung des Checkpoints von Punkt zu Punkt animiert ist.



### 5.22.129 chart.timeline.checkpointStyle.animationDuration

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Die Animationsdauer des Checkpoints in der Zeitstrahl-Komponente.

### 5.22.130 chart.timeline.checkpointStyle.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'quinticInOut' ]

Der Linderungseffekt der Checkpoint-Animation in der Zeitstrahl-Komponente.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn

- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 5.22.131 chart.timeline.controlStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Kontrollknöpfe, was beinhaltet: *Wiedergabe, Vorheriger* und *Nächster*.

### 5.22.132 chart.timeline.controlStyle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Kontrollknöpfe angezeigt werden sollen. False um alle zu verbergen.

### 5.22.133 chart.timeline.controlStyle.showPlayBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Wiedergabe-Knopf angezeigt werden soll.

### 5.22.134 chart.timeline.controlStyle.showPrevBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Vorheriger-Knopf angezeigt werden soll.

### 5.22.135 chart.timeline.controlStyle.showNextBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Nächster-Knopf angezeigt werden soll.

### 5.22.136 chart.timeline.controlStyle.itemSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 22 ]

Größe der Kontrollknöpfe, in Pixeln (px).

### 5.22.137 chart.timeline.controlStyle.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Intervall zwischen den Kontrollknöpfen, in Pixeln (px).

### 5.22.138 chart.timeline.controlStyle.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'left' ]

Die Platzierung der Kontrollknöpfe.

- Wenn `timeline.orient` auf 'horizontal' steht, sind 'left' und 'right' gültig.
- Wenn `timeline.orient` auf 'vertical' steht, sind 'top' und 'bottom' gültig.

Erlaubte Werte

- left
- right
- top
- bottom

### 5.22.139 `chart.timeline.controlStyle.playIcon`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Abspielen* Status des *Abspielen* Knopfs

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.22.140 `chart.timeline.controlStyle.stopIcon`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Stoppen* Status des *Abspielen* Knopfs.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.22.141 chart.timeline.controlStyle.prevlIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Vorheriger* Knopf.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.22.142 chart.timeline.controlStyle.nextIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Nächster* Knopf.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.22.143 chart.timeline.controlStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Knopffarbe.

### 5.22.144 chart.timeline.controlStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Farbe des Knopfrandes

### 5.22.145 chart.timeline.controlStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Knopfes.

### 5.22.146 chart.timeline.controlStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Knopfes im *hervorgehobenen* Zustand (wenn die Maus auf ihn zeigt).

### 5.22.147 chart.timeline.controlStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]



Knopffarbe.

### 5.22.148 chart.timeline.controlStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Knopfrandes.

### 5.22.149 chart.timeline.controlStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Breite des Knopfrandes.

## 5.23 ECharts (4) Title

### 5.23.1 chart.title.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titelkomponente, einschließlich Haupttitel und Untertitel.

In ECharts 2.x kann eine einzelne Instanz von Echarts maximal eine Titelkomponente enthalten. Von ECharts 3 kann es jedoch eine oder mehrere Titelkomponenten geben. Es ist möglich, dass alle benötigten Titel mehrerer Diagramme in einer Instanz definiert werden.

### 5.23.2 chart.title.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 5.23.3 chart.title.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Gibt an ob die Titelkomponente angezeigt wird.

### 5.23.4 chart.title.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Haupttitel-Text. Unterstützt \n für neue Zeilen.

### 5.23.5 chart.title.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Der Hyperlink des Haupttiteltextes.

### 5.23.6 chart.title.target

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Öffnen des Hyperlinks des Haupttitels in dem angegebenen Tab. Optionen:

- 'self' öffnet es in der aktuellen Tab.
- 'blank' öffnet es in einem neuen Tab.

### 5.23.7 chart.title.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil der Komponente.

### 5.23.8 chart.title.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#333" ]

Haupttitel Textfarbe.

### 5.23.9 chart.title.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Haupttitel Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 5.23.10 chart.title.textStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Haupttitel Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 5.23.11 chart.title.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Haupttitel Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.23.12 chart.title.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 18 ]

Haupttitel Schriftgröße.

### 5.23.13 chart.title.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### 5.23.14 chart.title.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'top'
- 'middle'
- 'bottom'

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### 5.23.15 chart.title.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.23.16 chart.title.textStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, standardmäßig transparent.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 5.23.17 chart.title.textStyle.borderColor

<b>Datentyp</b>	COLOR
-----------------	-------



<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Titels. Unterstützt das gleiche Farbformat wie backgroundColor.

### 5.23.18 chart.title.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Titels.

### 5.23.19 chart.title.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die vier Ecken gesondert zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // Setze den gleichen Radius für alle vier Ecken
borderRadius: [5, 5, 0, 0] // (Im Uhrzeigersinn oben links, oben rechts, unten rechts
und unten links)
```

### 5.23.20 chart.title.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 5.23.21 chart.title.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 5.23.22 chart.title.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 5.23.23 chart.title.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 5.23.24 chart.title.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 5.23.25 chart.title.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 5.23.26 chart.title.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.23.27 chart.title.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.23.28 chart.title.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.23.29 chart.title.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.23.30 chart.title.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.23.31 chart.title.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.23.32 chart.title.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.23.33 chart.title.textStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.23.34 chart.title.subtext

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Untertiteltext. Unterstützt \n für neue Zeilen.



### 5.23.35 chart.title.sublink

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '' ]

Der Hyperlink des Untertiteltexes.

### 5.23.36 chart.title.subtarget

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Öffnen des Hyperlinks des Untertitels in dem angegebenen Tab. Optionen:

- 'self' öffnet es in der aktuellen Tab.
- 'blank' öffnet es in einem neuen Tab.

### 5.23.37 chart.title.subtextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Stil des Subtexts.

### 5.23.38 chart.title.subtextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#aaa" ]

Untertitel Textfarbe.

### 5.23.39 chart.title.subtextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Untertitel Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 5.23.40 chart.title.subtextStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Untertitel Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 5.23.41 chart.title.subtextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Untertitel Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.23.42 chart.title.subtextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Untertitel Schriftgröße.

### 5.23.43 chart.title.subtextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 5.23.44 chart.title.subtextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'top'
- 'middle'
- 'bottom'

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### 5.23.45 chart.title.subtextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.23.46 chart.title.subtextStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 5.23.47 chart.title.subtextStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Subtexts. Unterstützt das gleiche Farbformat wie backgroundColor.

### 5.23.48 chart.title.subtextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Subtexts.

### 5.23.49 chart.title.subtextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 5.23.50 chart.title.subtextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Subtext um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.23.51 chart.title.subtextStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.23.52 chart.title.subtextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 5.23.53 chart.title.subtextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.23.54 chart.title.subtextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

### 5.23.55 chart.title.subtextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 5.23.56 chart.title.subtextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.23.57 chart.title.subtextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.23.58 chart.title.subtextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.23.59 chart.title.subtextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.23.60 chart.title.subtextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.23.61 chart.title.subtextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.23.62 chart.title.subtextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.23.63 chart.title.subtextStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 5.23.64 chart.title.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ob ein Event ausgelöst wird.

### 5.23.65 chart.title.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der Titel um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.23.66 chart.title.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Der Abstand zwischen dem Haupttitel und dem Untertitel.

### 5.23.67 chart.title.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

### 5.23.68 chart.title.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.23.69 chart.title.left

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

#### 5.23.70 chart.title.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.23.71 chart.title.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

## Erlaubte Werte

- {percent}
- {integer}

### 5.23.72 chart.title.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}

### 5.23.73 chart.title.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 5.23.74 chart.title.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Titels. Unterstützt das gleiche Farbformat wie backgroundColor.

### 5.23.75 chart.title.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Titels.

### 5.23.76 chart.title.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 5.23.77 chart.title.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 5.23.78 chart.title.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.23.79 chart.title.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 5.23.80 chart.title.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

## 5.24 ECharts (4) Toolbox

### 5.24.1 chart.toolbox.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Komponenten-ID, nicht spezifiziert per Default. Wenn eine ID gesetzt wird, kann auf die Komponente per ID über die EChart-JavaScript-API zugegriffen werden.

### 5.24.2 chart.toolbox.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Zeigt die Tool-Box-Komponente an.

### 5.24.3 chart.toolbox.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Layout-Orientierung der EChart-Toolbox Icons.

Erlaubte Werte

- horizontal
- vertical

### 5.24.4 chart.toolbox.itemSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Die Größe der EChart-Toolbox-Icons.

### 5.24.5 chart.toolbox.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Der Abstand zwischen den Toolbox-Icons. Bezieht sich je nach Layout auf den vertikalen oder horizontalen Abstand.

### 5.24.6 chart.toolbox.showTitle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Aktiviert die Titel-Text zu den Toolbox-Icons bei Mouse-Over.

true oder false

### 5.24.7 chart.toolbox.feature.saveAsImage

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Als Bild speichern.

#### 5.24.8 chart.toolbox.feature.saveAsImage.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'png' ]

Format in dem das Bild gespeichert wird. Unterstützt 'png' und 'jpeg'.

#### 5.24.9 chart.toolbox.feature.saveAsImage.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name mit dem das Bild gespeichert wird. Default-Wert ist title.text.

#### 5.24.10 chart.toolbox.feature.saveAsImage.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Hintergrundfarbe mit der das Bild gespeichert wird. Default-Wert ist backgroundColor. Ist backgroundColor nicht gesetzt, wird die Farbe Weiß verwendet.

### 5.24.11 chart.toolbox.feature.saveAsImage.excludeComponents

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['toolbox'] ]

Komponenten die beim Export ausgeschlossen werden. Per default wird toolbox ausgeschlossen.

### 5.24.12 chart.toolbox.feature.saveAsImage.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 5.24.13 chart.toolbox.feature.saveAsImage.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'save as image' ]

### 5.24.14 chart.toolbox.feature.saveAsImage.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.15 chart.toolbox.feature.saveAsImage.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellung des 'Als Bild speichern'-Icons.

### 5.24.16 chart.toolbox.feature.saveAsImage.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.17 chart.toolbox.feature.saveAsImage.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.24.18 chart.toolbox.feature.saveAsImage.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.19 chart.toolbox.feature.saveAsImage.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.20 chart.toolbox.feature.saveAsImage.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 5.24.21 chart.toolbox.feature.saveAsImage.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.24.22 chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.24.23 chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.24 chart.toolbox.feature.saveAsImage.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.25 chart.toolbox.feature.saveAsImage.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 5.24.26 chart.toolbox.feature.saveAsImage.iconStyle.textAlign

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 5.24.27 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.28 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 5.24.29 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.30 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.31 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 5.24.32 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.24.33 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.24.34 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.35 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.36 chart.toolbox.feature.saveAsImage.pixelRatio

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Auflösungsrate in der das Bild gespeichert wird, welche standardmäßig die des Containers ist. Werte größer als 1 (z.B. 2) werden unterstützt, falls eine höhere Auflösung benötigt wird.

### 5.24.37 chart.toolbox.feature.restore

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Einstellungen wiederherstellen"-Objekt

### 5.24.38 chart.toolbox.feature.restore.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 5.24.39 chart.toolbox.feature.restore.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'restore' ]

#### 5.24.40 chart.toolbox.feature.restore.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 5.24.41 chart.toolbox.feature.restore.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Wiederherstellen"-Icons

#### 5.24.42 chart.toolbox.feature.restore.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

#### 5.24.43 chart.toolbox.feature.restore.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

#### 5.24.44 chart.toolbox.feature.restore.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.45 chart.toolbox.feature.restore.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.46 chart.toolbox.feature.restore.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 5.24.47 chart.toolbox.feature.restore.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 5.24.48 chart.toolbox.feature.restore.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.24.49 chart.toolbox.feature.restore.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 5.24.50 chart.toolbox.feature.restore.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.24.51 chart.toolbox.feature.restore.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

#### 5.24.52 chart.toolbox.feature.restore.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

## Erlaubte Werte

- left
- right

## 5.24.53 chart.toolbox.feature.restore.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

## 5.24.54 chart.toolbox.feature.restore.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

## 5.24.55 chart.toolbox.feature.restore.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

#### 5.24.56 chart.toolbox.feature.restore.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.57 chart.toolbox.feature.restore.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

## 5.24.58 chart.toolbox.feature.restore.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

## 5.24.59 chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.24.60 chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.61 chart.toolbox.feature.restore.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.62 chart.toolbox.feature.dataView

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tool zur Dateneinsicht, welches Daten im momentanen Diagramm anzeigt und das Diagramm nach Änderungen erneuert.

### 5.24.63 chart.toolbox.feature.dataView.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

#### 5.24.64 chart.toolbox.feature.dataView.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'data view' ]

#### 5.24.65 chart.toolbox.feature.dataView.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 5.24.66 chart.toolbox.feature.dataView.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Dateneinsicht"-Icons

### 5.24.67 chart.toolbox.feature.dataView.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.68 chart.toolbox.feature.dataView.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.24.69 chart.toolbox.feature.dataView.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.70 chart.toolbox.feature.dataView.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.71 chart.toolbox.feature.dataView.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.72 chart.toolbox.feature.dataView.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 5.24.73 chart.toolbox.feature.dataView.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.24.74 chart.toolbox.feature.dataView.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.75 chart.toolbox.feature.dataView.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.76 chart.toolbox.feature.dataView.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

#### Erlaubte Werte

- left
- right
- top
- bottom

### 5.24.77 chart.toolbox.feature.dataView.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 5.24.78 chart.toolbox.feature.dataView.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolstil im hervorgehobenen Zustand.

### 5.24.79 chart.toolbox.feature.dataView.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

## 5.24.80 chart.toolbox.feature.dataView.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

## 5.24.81 chart.toolbox.feature.dataView.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

## 5.24.82 chart.toolbox.feature.dataView.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.83 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 5.24.84 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

## 5.24.85 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.24.86 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 5.24.87 chart.toolbox.feature.dataView.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.88 chart.toolbox.feature.dataView.readOnly

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ob es nur Lesezugriff erlaubt.

### 5.24.89 chart.toolbox.feature.dataView.optionToContent

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

```
(option:Object) => HTMLDomElement|string
```

Definiere eine Funktion, um `dataView` zu präsentieren. Sie wird verwendet, um die Standard-textarea für ergiebigere Datenverarbeitung zu ersetzen. Sie kann ein DOM-Objekt oder einen HTML-String zurückgeben.

Zum Beispiel:

```

optionToContent: function(opt) {
  var axisData = opt.xAxis[0].data;
  var series = opt.series;
  var table = '<table style="width:100%;text-align:center"><tbody><tr>'
    + '<td>Time:</td>'
    + '<td>' + series[0].name + '</td>'
    + '<td>' + series[1].name + '</td>'
    + '</tr>';
  for (var i = 0, l = axisData.length; i < l; i++) {
    table += '<tr>'
      + '<td>' + axisData[i] + '</td>'
      + '<td>' + series[0].data[i] + '</td>'
      + '<td>' + series[1].data[i] + '</td>'
      + '</tr>';
  }
  table += '</tbody></table>';
  return table;
}

```

### 5.24.90 chart.toolbox.feature.dataView.contentToOption

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

```
(container:HTMLDomElement, option:Object) => Object
```

Wenn `optionToContent` verwendet wird, falls die Aktualisierung von Diagrammen nach Datenveränderungen unterstützt werden soll, muss die Logik der Zusammenführung von Optionen in dieser Funktion implementiert werden.

### 5.24.91 chart.toolbox.feature.dataView.lang

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['data view', 'turn off', 'refresh']]

Es gibt 3 Namen im data view, diese sind 'data view', 'turn off' und 'refresh'.

#### 5.24.92 chart.toolbox.feature.dataView.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Hintergrundfarbe der schwebenden Ebene im data view.

#### 5.24.93 chart.toolbox.feature.dataView.textareaColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Hintergrundfarbe des Eingabebereichs der schwebenden Ebene im data view.

#### 5.24.94 chart.toolbox.feature.dataView.textareaBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Rahmenfarbe des Eingabebereichs der schwebenden Ebene im data view.

### 5.24.95 chart.toolbox.feature.dataView.textColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Textfarbe.

### 5.24.96 chart.toolbox.feature.dataView.buttonColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Knopffarbe.

### 5.24.97 chart.toolbox.feature.dataView.buttonTextColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Farbe der Knopfbeschriftung.

### 5.24.98 chart.toolbox.feature.dataZoom

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Datenbereich-zooming, welches momentan nur rechteckige Koordinaten unterstützt.

### 5.24.99 chart.toolbox.feature.dataZoom.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

## 5.24.100 chart.toolbox.feature.dataZoom.title

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wiederhergestellter und gezoomter Titeltext.

## 5.24.101 chart.toolbox.feature.dataZoom.title.zoom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'area zooming' ]

## 5.24.102 chart.toolbox.feature.dataZoom.title.back

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'restore area zooming' ]

### 5.24.103 chart.toolbox.feature.dataZoom.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Zooming und Wiederherstellung"-Icon Pfad

### 5.24.104 chart.toolbox.feature.dataZoom.icon.zoom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.105 chart.toolbox.feature.dataZoom.icon.back

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.106 chart.toolbox.feature.dataZoom.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Zoom auf Datenbereich"-Icons

### 5.24.107 chart.toolbox.feature.dataZoom.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.108 chart.toolbox.feature.dataZoom.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.24.109 chart.toolbox.feature.dataZoom.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.110 chart.toolbox.feature.dataZoom.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.111 chart.toolbox.feature.dataZoom.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.112 chart.toolbox.feature.dataZoom.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 5.24.113 chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.24.114 chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 5.24.115 chart.toolbox.feature.dataZoom.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.24.116 chart.toolbox.feature.dataZoom.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 5.24.117 chart.toolbox.feature.dataZoom.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 5.24.118 chart.toolbox.feature.dataZoom.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.119 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 5.24.120 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.121 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.122 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.123 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.24.124 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.24.125 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.126 chart.toolbox.feature.dataZoom.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.127 chart.toolbox.feature.dataZoom.xAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert welche xAxis kontrolliert werden sollen. Standardmäßig werden alle X-Achsen kontrolliert. Wird der Wert auf false gesetzt, werden keine X-Achsen kontrolliert. Wird der Wert auf 3 gesetzt, wird die X-Achse mit axisIndex von 3 kontrolliert. Wird der Wert auf [0, 3] gesetzt, werden die X-Achsen mit axisIndex von 0 und 3 kontrolliert.

### 5.24.128 chart.toolbox.feature.dataZoom.yAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert welche yAxis kontrolliert werden sollen. Standardmäßig werden alle Y-Achsen kontrolliert. Wird der Wert auf false gesetzt, werden keine Y-Achsen kontrolliert. Wird der Wert auf 3 gesetzt, wird die Y-Achse mit axisIndex von 3 kontrolliert. Wird der Wert auf [0, 3] gesetzt, werden die Y-Achsen mit axisIndex von 0 und 3 kontrolliert.

### 5.24.129 chart.toolbox.feature.magicType

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Magic type wechsel. Beispiel:

```
feature: {
  magicType: {
    type: ['line', 'bar', 'stack', 'tiled']
  }
}
```

### 5.24.130 chart.toolbox.feature.magicType.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das tool angezeigt werden soll.

### 5.24.131 chart.toolbox.feature.magicType.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Aktivierte magic types. Möglich sind 'line' (für Liniendiagramme), 'bar' (für Balkendiagramme), 'stack' (für gestapelte Diagramme) und 'tiled' (für gekachelte Diagramme).

### 5.24.132 chart.toolbox.feature.magicType.title

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titel für verschiedene typen, können separat konfiguriert werden.

### 5.24.133 chart.toolbox.feature.magicType.title.line

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for line charts' ]

### 5.24.134 chart.toolbox.feature.magicType.title.bar

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for bar charts' ]



### 5.24.135 chart.toolbox.feature.magicType.title.stack

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for stacked charts' ]

### 5.24.136 chart.toolbox.feature.magicType.title.tiled

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for tiled charts' ]

### 5.24.137 chart.toolbox.feature.magicType.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die verschiedenen Typen von icon-Pfad, welche individuell konfiguriert werden können.

### 5.24.138 chart.toolbox.feature.magicType.icon.line

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.139 chart.toolbox.feature.magicType.icon.bar

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.140 chart.toolbox.feature.magicType.icon.stack

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 5.24.141 chart.toolbox.feature.magicType.icon.tiled

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 5.24.142 chart.toolbox.feature.magicType.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Magic-Typ wechseln"-Icons

### 5.24.143 chart.toolbox.feature.magicType.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.144 chart.toolbox.feature.magicType.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.24.145 chart.toolbox.feature.magicType.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

#### 5.24.146 chart.toolbox.feature.magicType.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.147 chart.toolbox.feature.magicType.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.148 chart.toolbox.feature.magicType.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 5.24.149 chart.toolbox.feature.magicType.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.24.150 chart.toolbox.feature.magicType.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.151 chart.toolbox.feature.magicType.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.152 chart.toolbox.feature.magicType.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

#### Erlaubte Werte

- left
- right
- top
- bottom

### 5.24.153 chart.toolbox.feature.magicType.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 5.24.154 chart.toolbox.feature.magicType.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolstil im hervorgehobenen Zustand.

### 5.24.155 chart.toolbox.feature.magicType.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.156 chart.toolbox.feature.magicType.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 5.24.157 chart.toolbox.feature.magicType.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.158 chart.toolbox.feature.magicType.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.159 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.160 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

## 5.24.161 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 5.24.162 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 5.24.163 chart.toolbox.feature.magicType.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.164 chart.toolbox.feature.magicType.option

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationsobjekt für jeden Typ, welche verwendet werden wenn zu einem bestimmten Typ gewechselt wird..

### 5.24.165 chart.toolbox.feature.magicType.option.line

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 5.24.166 chart.toolbox.feature.magicType.option.bar

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 5.24.167 chart.toolbox.feature.magicType.option.stack

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 5.24.168 chart.toolbox.feature.magicType.option.tiled

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 5.24.169 chart.toolbox.feature.magicType.seriesIndex

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Serienliste für jeden Typ.

### 5.24.170 chart.toolbox.feature.magicType.seriesIndex.line

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 5.24.171 chart.toolbox.feature.magicType.seriesIndex.bar

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 5.24.172 chart.toolbox.feature.magicType.seriesIndex.stack

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 5.24.173 chart.toolbox.feature.magicType.seriesIndex.tiled

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 5.24.174 chart.toolbox.feature.brush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Pinsel-Auswählen Icon.

Kann auch bei brush.toolbox konfiguriert werden.

## 5.24.175 chart.toolbox.feature.brush.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das tool angezeigt werden soll.

## 5.24.176 chart.toolbox.feature.brush.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verwendete Icons, deren Werte sind:

- 'rect': Ermöglicht Auswahl mit rechteckigem Bereich.
- 'polygon': Ermöglicht Auswahl mit beliebiger Form.
- 'lineX': Ermöglicht horizontale Auswahl.
- 'lineY': Ermöglicht vertikale Auswahl.
- 'keep': Wechsel zwischen *einzelner Auswahl* und *multipler Auswahl*. Die Letztere kann mehrere Bereiche auswählen während die Erstere vorherige Auswahlen abbricht.
- 'clear': Bricht alle Auswahlen ab.

## 5.24.177 chart.toolbox.feature.brush.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon-Pfad für jedes Icon.

## 5.24.178 chart.toolbox.feature.brush.icon.rect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.179 chart.toolbox.feature.brush.icon.polygon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.180 chart.toolbox.feature.brush.icon.lineX

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.181 chart.toolbox.feature.brush.icon.lineY

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.182 chart.toolbox.feature.brush.icon.keep

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.183 chart.toolbox.feature.brush.icon.clear

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 5.24.184 chart.toolbox.feature.brush.title

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titel.

### 5.24.185 chart.toolbox.feature.brush.title.rect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Rectangle selection' ]

#### 5.24.186 chart.toolbox.feature.brush.title.polygon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Polygon selection' ]

#### 5.24.187 chart.toolbox.feature.brush.title.lineX

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Horizontal selection' ]

#### 5.24.188 chart.toolbox.feature.brush.title.lineY

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Vertical selection' ]

### 5.24.189 chart.toolbox.feature.brush.title.keep

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Keep previous selection' ]

### 5.24.190 chart.toolbox.feature.brush.title.clear

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Clear selection' ]

### 5.24.191 chart.toolbox.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des geteilten Icons.

### 5.24.192 chart.toolbox.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.193 chart.toolbox.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 5.24.194 chart.toolbox.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.195 chart.toolbox.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 5.24.196 chart.toolbox.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.197 chart.toolbox.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 5.24.198 chart.toolbox.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 5.24.199 chart.toolbox.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.200 chart.toolbox.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.201 chart.toolbox.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

#### Erlaubte Werte

- left
- right
- top
- bottom

### 5.24.202 chart.toolbox.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 5.24.203 chart.toolbox.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolstil im hervorgehobenen Zustand.

### 5.24.204 chart.toolbox.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 5.24.205 chart.toolbox.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 5.24.206 chart.toolbox.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 5.24.207 chart.toolbox.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 5.24.208 chart.toolbox.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 5.24.209 chart.toolbox.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 5.24.210 chart.toolbox.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.24.211 chart.toolbox.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.24.212 chart.toolbox.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.24.213 chart.toolbox.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

### 5.24.214 chart.toolbox.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.24.215 chart.toolbox.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 5.24.216 chart.toolbox.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 5.24.217 chart.toolbox.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

### 5.24.218 chart.toolbox.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der unteren Seite des Containers.



Der Wert von `bottom` kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

### 5.24.219 `chart.toolbox.width`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Toolbox-Komponente. Standardmäßig adaptiv.

### 5.24.220 `chart.toolbox.height`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Toolbox-Komponente. Standardmäßig adaptiv.

### 5.24.221 `chart.toolbox.defaulticons`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die standardmäßigen Icons für die Toolbox verwendet werden sollen, oder die von Tradui entworfenen Icons.

### 5.24.222 chart.toolbox.feature.myToolToggleSeriesVisibility.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob die Toolbox-Option 'ToggleVisibility' angezeigt werden soll. Wird dieser Knopf angeklickt, so werden alle Legendeninformationen sichtbar oder unsichtbar

### 5.24.223 chart.toolbox.feature.myToolToggleSeriesVisibility\_RescaleYAxis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob die Toolbox-Option für 'ToggleVisibility' angezeigt werden soll. Wird dieser Knopf angeklickt, so werden alle Legendeninformationen sichtbar oder unsichtbar. Zusätzlich kann die Skalierung der Y-Achse durch Setzen des Wertes mit der Option "`chart.toolbox.feature.Rescale Axis.function`" angepasst werden.

### 5.24.224 chart.toolbox.feature.RescaleYAxis.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Skalierung der Y-Achse kann mit dieser Option für die Toolbox-Option 'ToggleVisibility' angepasst werden.

### 5.24.225 chart.toolbox.feature.myToolRestore.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Gibt an, ob die benutzerdefinierte Wiederherstellungsoption angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 5.24.226 chart.toolbox.feature.myToolBarChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte BarChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 5.24.227 chart.toolbox.feature.myToolLineChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte LineChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 5.24.228 chart.toolbox.feature.myToolStackedChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte StackedChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 5.24.229 chart.toolbox.feature.myToolRestore.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die Restore-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("chart.toolbox.feature.myToolRestore.additionalscript", "console.log('here');");</span>
```

### 5.24.230 chart.toolbox.feature.myToolBarChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die BarChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolBarChart.additionalscript", "console.log('here');");
```

### 5.24.231 chart.toolbox.feature.myToolLineChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13

### Schlagworte / Tags

Ein customized Javascript-Code kann für die LineChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolLineChart.additionalscript",
"console.log('here');");
```

## 5.24.232 chart.toolbox.feature.myToolStackedChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die StackedChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolStackedChart.additionalscript",
"console.log('here');");
```

## 5.24.233 chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die ToggleVisibility-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption(
  "chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript",
  "console.log('here');");
```

### 5.24.234 chart.toolbox.feature.myToolToggleSeriesVisibility\_RescaleYAxis.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die ToggleVisibility-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption(
  "chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.additionalscript",
  "console.log('here');");
```

## 5.25 ECharts (4) Tooltip

### 5.25.1 chart.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip Komponente.

### Allgemeine Einführung:

Tooltips können an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

## 5.25.2 chart.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

## 5.25.3 chart.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'



Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.

ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentypen in grid und polar unterstützt. Weiterhin lassen sich Achsen über `axisPointer.axis` zuweisen.

- 'none'  
Durch nichts ausgelöst.

## Erlaubte Werte

- item
- axis
- none

### 5.25.4 chart.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

`tooltip.axisPointer` ist wie syntaktischer Zucker von `axisPointer`-Einstellungen an Achsen (zum Beispiel `xAxis.axisPointer` oder `angleAxis.axisPointer`). Detailliertere Einstellungen können an `someAxis.axisPointer` vorgenommen werden. Im Normalfall ist die Verwendung von `tooltip.axisPointer` jedoch praktischer.

**Hinweis:** Konfigurationen an `tooltip.axisPointer` haben eine niedrigere Priorität als `someAxis.axisPointer`

### 5.25.5 chart.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

Erlaubte Werte

- line
- shadow
- none
- cross

### 5.25.6 chart.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

Erlaubte Werte

- auto
- x
- y
- radius
- angle

### 5.25.7 chart.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 5.25.8 chart.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 5.25.9 chart.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Label des axisPointers.

### 5.25.10 chart.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 5.25.11 chart.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 5.25.12 chart.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

**Rückgabewert:**

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 5.25.13 chart.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 5.25.14 chart.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.25.15 chart.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 5.25.16 chart.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.25.17 chart.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.25.18 chart.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße

### 5.25.19 chart.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 5.25.20 chart.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 5.25.21 chart.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.25.22 chart.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.25.23 chart.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.25.24 chart.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.25.25 chart.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.25.26 chart.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.25.27 chart.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.25.28 chart.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```

// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]

```

### 5.25.29 chart.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 5.25.30 chart.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 5.25.31 chart.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 5.25.32 chart.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.25.33 chart.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.25.34 chart.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.25.35 chart.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.25.36 chart.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'line' ist.

### 5.25.37 chart.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.25.38 chart.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite

## 5.25.39 chart.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp

Erlaubte Werte

- solid
- dashed
- dotted

## 5.25.40 chart.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.25.41 chart.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.25.42 chart.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.25.43 chart.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 5.25.44 chart.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 5.25.45 chart.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

#### 5.25.46 chart.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.25.47 chart.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.25.48 chart.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.25.49 chart.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.25.50 chart.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.25.51 chart.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.25.52 chart.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.



### 5.25.53 chart.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 5.25.54 chart.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 5.25.55 chart.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

## 5.25.56 chart.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 5.25.57 chart.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 5.25.58 chart.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 5.25.59 chart.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 5.25.60 chart.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 5.25.61 chart.tooltip.showContent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der schwebende Tooltip-Layer angezeigt werden soll. Dieser Wert sollte auf false gesetzt werden, falls der Tooltip nur sein Event auslösen oder den axisPointer ohne Inhalt anzeigen soll.

### 5.25.62 chart.tooltip.alwaysShowContent

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Inhalt des Tooltips durchgehend angezeigt werden soll. Standardmäßig wird er nach einiger Zeit verborgen. Dieser Wert kann auf true gesetzt werden, um das Anzeigen zu bewahren.

Dieses Attribut wurde zu ECharts 3.0 neu hinzugefügt.

### 5.25.63 chart.tooltip.triggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mousemove|click' ]

Konditionen unter denen der Tooltip auslöst. Mögliche Werte:

- 'mousemove'  
Löse aus wenn die Maus sich bewegt.
- 'click'  
Löse aus wenn die Maus klickt.
- 'mousemove|click'  
Löse aus wenn die Maus klickt und sich bewegt.
- 'none'  
Löse nicht durch bewegen und klicken der Maus aus. Der Tooltip kann manuell ausgelöst und verborgen werden indem action.tooltip.showTip und action.tooltip.hideTip aufgerufen werden. In diesem Fall kann er auch durch axisPointer.handle ausgelöst werden.

#### Erlaubte Werte

- mousemove
- click
- none
- mousemove|click

### 5.25.64 chart.tooltip.showDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Zeitverzögerung in ms bis der Tooltip angezeigt wird. Standardmäßig gibt es keine Verzögerung, und das setzen einer solchen ist nicht empfohlen. Nur gültig wenn triggerOn auf 'mousemove' gesetzt ist.

### 5.25.65 chart.tooltip.hideDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Verzögerung in ms bis der Tooltip verborgen wird. Ist ungültig wenn alwaysShowContent auf true gesetzt ist.

### 5.25.66 chart.tooltip.enterable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Maus den schwebenden Tooltip-Layer betreten darf. Muss mit Inhalten des Tooltips, wie Links oder Schaltern, interagiert werden, so kann dieser Wert auf true gesetzt werden.

### 5.25.67 chart.tooltip.renderMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'html' ]

Rendermodus für den Tooltip. Steht standardmäßig auf 'html', wodurch ein zusätzliches DOM-Element für den Tooltip verwendet wird. Kann auch auf 'richText' gesetzt werden, wodurch der Tooltip im Canvas-Element gerendert wird (SVG rich text ist noch nicht implementiert). Dies ist sehr nützlich für Umgebungen die nicht über DOM verfügen, wie zum Beispiel Wechat Applikationen.

#### Erlaubte Werte

- html
- richText

### 5.25.68 chart.tooltip.confine

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Inhalt des Tooltips durch das rechteckige Fenster des Diagramms begrenzt werden soll.

Nützlich wenn der Tooltip abgeschnitten wird weil 'overflow: hidden' auf dem äußeren DOM der Diagramm-Instanz gesetzt ist oder wegen kleinen Bildschirmen auf Mobilgeräten.



### 5.25.69 chart.tooltip.transitionDuration

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.4 ]

Die Übergangsdauer der Animation des Tooltips in Sekunden. Wird der Wert auf 0 gesetzt, so bewegt sich der Tooltip eng mit der Maus.

### 5.25.70 chart.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLElement, rect: Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 5.25.71 chart.tooltip.formatter

**Datentyp**

UNKNOWN

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 5.25.72 chart.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 5.25.73 chart.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 5.25.74 chart.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 5.25.75 chart.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.25.76 chart.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 5.25.77 chart.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 5.25.78 chart.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.25.79 chart.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

## 5.25.80 chart.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.25.81 chart.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 5.25.82 chart.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 5.25.83 chart.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.25.84 chart.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.25.85 chart.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.25.86 chart.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.25.87 chart.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.25.88 chart.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.25.89 chart.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.25.90 chart.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.25.91 chart.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

## 5.26 ECharts (4) Visual Map

### 5.26.1 chart.visualMap.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

## Schlagworte / Tags

visualMap ist eine Art Komponente für visuelles Encoding, welche die Daten auf visuelle Kanäle mappt. Zu diesen Kanälen zählt:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbton in HSL.

Mehrere visualMap-Komponenten können in einer Diagramm-Instanz definiert werden, wodurch verschiedene Dimensionen einer Datenserie auf verschiedene visuelle Kanäle gemappt werden können.

visualMap kann als stetig oder stückweise definiert werden, was über die Eigenschaft type geschieht. Zum Beispiel:

```
option = {
  visualMap: [
    { // Die erste visualMap-Komponente
      type: 'continuous', // Definiert als stetige visualMap
      ...
    },
    { // Die zweite visualMap-Komponente
      type: 'piecewise', // Definiert als stückweise visualMap
      ...
    }
  ],
  ...
};
```

### Configure mapping

Die Dimension von series.data kann mit visualMap.dimension spezifiziert werden, von wo aus der Wert geholt und auf visuelle Kanäle gemappt werden kann. Diese Kanäle können in visualMap.inRange und visualMap.outOfRange definiert werden.

Soll in einer von einer visualMap kontrollierten Serie ein Element dieser Kontrolle entkommen, kann es wie folgt gesetzt werden:

```

series: {
  type: '...',
  data: [
    {name: 'Shanghai', value: 251},
    {name: 'Haikou', value: 21},
    // Markieren als `visualMap: false`, wodurch dieses Element nicht mehr von
    // visualMap kontrolliert wird
    // und die visuelle Konfiguration der Serie (wie color, symbol, ...) auf das
    // Element angewendet werden kann.
    {name: 'Beijing', value: 821, visualMap: false},
    ...
  ]
}

```

### Die Beziehung zwischen visualMap von ECharts3 und dataRange von ECharts2

dataRange aus ECharts2 wurde zu visualMap umbenannt, und das Ausmaß der Funktionalitäten wurde stark erweitert. Die Konfigurationen von dataRange sind kompatibel mit ECharts3, wo diese automatisch zu visualMap umgewandelt werden. Es wird empfohlen, in ECharts 3 visualMap statt dataRange zu verwenden.

#### 5.26.2 chart.visualMap.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den Typ der visualMap.

#### Erlaubte Werte

- continuous
- piecewise

#### 5.26.3 chart.visualMap.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

#### 5.26.4 chart.visualMap.min

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert den minimalen Datenwert für die visualMap-Komponente. [visualMap.min, visualMap.max] stellen den Bereich des visuellen Mappings dar.

Es ist anzumerken, dass min und max explizit spezifiziert werden sollten, und standardmäßig [0, 200] sind, nicht etwa dataMin und dataMax aus series.data.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) müssen max und min nicht spezifiziert werden.

#### 5.26.5 chart.visualMap.max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 200 ]



Spezifiziert den maximalen Datenwert für die visualMap-Komponente. [visualMap.min, visualMap.max] stellen den Bereich des visuellen Mappings dar.

Es ist anzumerken, dass min und max explizit spezifiziert werden sollten, und standardmäßig [0, 200] sind, nicht etwa dataMin und dataMax aus series.data.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) müssen max und min nicht spezifiziert werden.

## 5.26.6 chart.visualMap.range

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den ausgewählten Bereich, also die Datenwerte, die zu den beiden Handles gehören. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    min: 0,
    max: 100,
    // Datenwerte die zu den beiden Handles gehören.
    range: [4, 15],
    ...
  }
});
```

### Automatische Anpassung wenn min oder max durch setOption verändert wird

- Wenn range nicht gesetzt ist (oder auf null oder undefined gesetzt ist)

Zum Beispiel:

```
chart.setOption({visualMap: {min: 10, max: 300}}); // range ist nicht gesetzt, wird
standardmäßig [min, max], also [10, 300].
```

```
chart.setOption({visualMap: {min: 0, max: 400}}); // Modifiziere min und max wieder
mit setOption.
// range wird dann automatisch modifiziert auf das neue [min, max], also [0, 400].
```

- Wenn range explizit gesetzt ist, zum Beispiel auf [10, 300]

```
Zum Beispiel:
chart.setOption({visualMap: {min: 10, max: 300, range: [20, 80]}}); // range wurde
auf [20, 80] gesetzt.

chart.setOption({visualMap: {min: 0, max: 400}}); // Modifiziere min und max wieder
mit setOption.
// range behält den originalen Wert ([20, 80]) und passt sich nicht automatisch an.

chart.setOption({visualMap: {range: null}}); // Setze range auf null
// Die automatische Anpassung von range aktiviert sich und setzt range auf [min,
max], also [0, 400].
```

Die range die von `getOption` zurückgegeben wird ist immer ein Array, aber nie null oder undefined.

### 5.26.7 chart.visualMap.calculable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Handles angezeigt werden sollen, welche gezogen werden können um den "ausgewählten Bereich" zu verändern.

Anmerkung: Um mit ECharts 2 kompatibel zu bleiben, bleibt die folgende, seltsam erscheinende Regel erhalten: Wenn `visualMap.type` nicht gesetzt ist und `visualMap.calculable` auf `true` gesetzt ist, wird `visualMap.type` automatisch auf `'continuous'` gesetzt. Einstellungen wie `visualMap.splitNumber` werden dabei ignoriert. Daher wird empfohlen, `visualMap.type` explizit zu setzen, um Unklarheiten vorzubeugen.

### 5.26.8 chart.visualMap.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert werden soll, wenn ein Handle gezogen wird.

- Für true wird die Ansicht in Echtzeit beim Ziehen aktualisiert.
- Für false wird die Ansicht aktualisiert, wenn das Ziehen beendet wurde.

## 5.26.9 chart.visualMap.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Layout der visualMap-Komponente umgekehrt werden soll.

### Für stetige visualMap:

Steht inverse auf false, so ist die Richtung des Layouts wie im kartesischen Koordinatensystem. Das heißt:

- Steht visualMap.orient auf 'vertical', so werden große Datenwerte oben und kleine Datenwerte unten platziert.
- Steht visualMap.orient auf 'horizontal', so werden große Datenwerte rechts und kleine Datenwerte links platziert.

### Für stückweise visualMap:

Im **CONTINUOUS-AVERAGE** Modus (wenn visualMap.splitNumber benutzt wird) ist die Regel des Layouts identisch zur stetigen visualMap.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) wird das Layout der Elemente durch ihre Reihenfolge in der Definition der Stücke bzw. Kategorien bestimmt. Das heißt:

- Steht inverse auf false:
  - Wenn visualMap.orient auf 'vertical' steht, wird pieces[0] bzw. categories[0] oben platziert.
  - Wenn visualMap.orient auf 'horizontal' steht, wird pieces[0] bzw. categories[0] links platziert.

Steht inverse auf true, so sind die Ergebnisse jeweils umgekehrt.

## 5.26.10 chart.visualMap.precision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die dezimale Präzision der Beschriftung, standardmäßig 0 (keine Dezimalstellen).

Im **CONTINUOUS-AVERAGE** Modus (wenn `visualMap.splitNumber` benutzt wird) ist die Regel des Layouts identisch zu `visualMap.inverse`, die Dezimalstellen passen sich automatisch an `series.data` an.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn `visualMap.pieces` benutzt wird) und im **CATEGORY** Modus (wenn `visualMap.categories` benutzt wird) ist die dezimale Präzision standardmäßig 0 (keine Dezimalstellen).

### 5.26.11 `chart.visualMap.itemWidth`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Breite der Hauptleiste der stetigen `visualMap`-Komponente, oder die Breite jedes grafischen Elements, welches ein Stück in der stückweisen `visualMap`-Komponente repräsentiert.

### 5.26.12 `chart.visualMap.itemHeight`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Höhe der Hauptleiste der stetigen `visualMap`-Komponente, oder die Höhe jedes grafischen Elements, welches ein Stück in der stückweisen `visualMap`-Komponente repräsentiert.

### 5.26.13 chart.visualMap.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Spezifiziert die Positionierung von Handles und Beschriftungen an der Hauptleiste. Mögliche Werte sind:

- 'auto': Automatisch entscheiden.
- 'left': Die Handles und Beschriftungen stehen auf der rechten Seite. Gültig, wenn orient auf 'vertical' steht.
- 'right': Die Handles und Beschriftungen stehen auf der linken Seite. Gültig, wenn orient auf 'vertical' steht.
- 'top': Die Handles und Beschriftungen stehen auf der unteren Seite. Gültig, wenn orient auf 'horizontal' steht.
- 'bottom': Die Handles und Beschriftungen stehen auf der oberen Seite. Gültig, wenn orient auf 'horizontal' steht.

#### Erlaubte Werte

- auto
- left
- right
- top
- bottom

### 5.26.14 chart.visualMap.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Beschriftung an beiden Enden der Skala, zum Beispiel ['High', 'Low'].

Die Regel, dass Beschriftungen nicht angezeigt werden wenn text verwendet wird, wird für Kompatibilität mit ECharts2 erhalten.

### 5.26.15 chart.visualMap.textGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

#### Für stetige visualMap

Die Distanz zwischen den Enden der Hauptleiste und den Beschriftungen. Einheit ist px. Siehe visualMap.text.

#### Für stückweise visualMap

Die Distanz zwischen den Enden der grafischen Elemente der Stücke und den Beschriftungen. Einheit ist px. Siehe visualMap.text.

### 5.26.16 chart.visualMap.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die visualMap-Komponente angezeigt werden soll. Steht der Wert auf false, wird die Komponente nicht angezeigt, kann aber immer noch visuelles Mapping von Datenwerten zu visuellen Kanälen im Diagramm durchführen.

### 5.26.17 chart.visualMap.dimension

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, welche Dimension verwendet werden soll, um Datenwerte aus series.data zu holen, die dann auf die visuellen Kanäle gemappt werden.

series.data kann als zweidimensionales Array betrachtet werden, zum Beispiel:

```
[
  [12, 23, 43],
  [12, 23, 43],
  [43, 545, 65],
  [92, 23, 33]
]
```

Jede Spalte des obigen Arrays wird als Dimension betrachtet. Wird visualMap.dimension zum Beispiel auf 1 gesetzt, so wird die zweite Spalte (23, 23, 545, 23) für das visuelle Mapping ausgewählt.

Verwendet standardmäßig die letzte Dimension der Daten.

### 5.26.18 chart.visualMap.seriesIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, auf welchen Serien visuelles Mapping durchgeführt werden soll, von welchen Serien Daten geholt werden.

Standardmäßig werden alle Serien verwendet.

### 5.26.19 chart.visualMap.hoverLink

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

hoverLink aktiviert das Hervorheben bestimmter grafischer Elemente des Diagramms wenn die Maus auf einen Punkt der visualMap zeigt, der über visuelles Mapping mit diesen grafischen Elementen verbunden ist.

Im umgekehrten Fall wird, wenn die Maus auf ein grafisches Element des Diagramms zeigt, dessen Wertbeschriftung an seiner verbundenen Stelle in der visualMap angezeigt.

### 5.26.20 chart.visualMap.inRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

inRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:



```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
Komponente selbst:  
    inRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle nur für die Zielserie.  
    target: {  
      inRange: {  
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
        symbolSize: [60, 200]  
      }  
    },  
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.  
    controller: {  
      inRange: {  
        symbolSize: [30, 100]  
      }  
    }  
  }  
]
```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    Komponente selbst:
    inRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in inRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus series.data) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von [visualMap.min, visualMap.max] in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei [visualMap.min, visualMap.max] auf [0, 100] gesetzt, und sei series.data [50, 10, 100]. Diese Werte sollen auf den opacity-Bereich [0.4, 1] gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die visualMap-Komponente berechnet dann linear die opacity-Werte [0.7, 0.44, 1] für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel opacity: [1, 0.4], wodurch das Endergebnis für die obige Serie [0.7, 0.96, 0.4] beträgt.

Hinweis: [visualMap.min, visualMap.max] sollte manuell gesetzt werden und ist standardmäßig [0, 200], nicht dataMin und dataMax aus series.data.

Wie muss die visualMap-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige visualMap wird verwendet, oder
- Eine stückweise visualMap wird verwendet und visualMap.categories ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundsätzlich wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. color: ['#333', '#777'].

- Einzelne Zahlen oder Strings können auch verwendet werden, die visualMap-Komponente wandelt solche in ein Array um. Zum Beispiel: opacity: 0.4 wird konvertiert zu opacity: [0.4, 0.4], color: '#333' wird konvertiert zu color: ['#333', '#333'].
- Für die visuellen Kanäle symbolSize, opacity, colorAlpha, colorLightness, colorSaturation und colorHue hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von visualMap.min, visueller Wert von visualMap.max]. Zum Beispiel: colorLightness: [0.8, 0.2] bedeutet, dass ein Datenwert in series.data der gleich visualMap.min ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich visualMap.max ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen [visualMap.min, visualMap.max] und [0.8, 0.2] gemappt.
- Für den visuellen Kanal color wird ein Array wie ['#333', '#78ab23', 'blue'] verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich visualMap.min sind, auf '#333' gemappt, Datenwerte, die gleich visualMap.max sind, auf 'blue' gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal symbol wird ein Array wie ['circle', 'rect', 'diamond'] verwendet, mit dem Datenwerte, die gleich visualMap.min sind, auf 'circle' gemappt werden, Datenwerte, die gleich visualMap.max sind, auf 'diamond' gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu visualMap.min und visualMap.max auf 'circle', 'rect' oder 'diamond' gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- opacity, colorAlpha, colorLightness, colorSaturation  
Die möglichen Werte liegen in [0, 1].
- colorHue  
Die möglichen Werte liegen in [0, 360].
- color  
Mögliche Werte sind RGB-Ausdrücke, wie 'rgb(128, 128, 128)', RGBA-Ausdrücke, wie 'rgba(128, 128, 128, 0.5)', oder Hex-Ausdrücke, wie '#ccc'.
- symbol  
Von ECharts bereitgestellte Icon-Typen sind 'circle', 'rect', 'roundRect', 'triangle', 'diamond', 'pin', 'arrow' und 'none'.  
Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.  
Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```

visualMap: {
  type: 'piecewise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  inRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
  Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}

```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem `setOption` aufgerufen wurde um die initiale Option zu setzen), kann `setOption` erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```

chart.setOption({
  visualMap: {
    inRange: {color: ['red', 'blue']}
  }
});

```

#### Hinweis:

- Diese `visualMap`-Eigenschaften (`inRange`, `outOfRange`, `target`, `controller`) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn `setOption` erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an `setOption` übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird `getOption()` -> modifiziere die erhaltene Option -> `setOption(modifizierte Option)`, strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.inRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.inRange.color = ['red', 'blue'];
option.visualMap.controller.inRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

**Hinweis:** Es gibt Standardfarben ['#f6efa6', '#d88273', '#bf444c'] in inRange wenn inRange nicht gesetzt wird. Sind diese unerwünscht, können sie durch inRange: { color: null } deaktiviert werden.

### 5.26.21 chart.visualMap.outOfRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

outOfRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    outOfRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle nur für die Zielserie.  
    target: {  
      outOfRange: {  
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
        symbolSize: [60, 200]  
      }  
    },  
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.  
    controller: {  
      outOfRange: {  
        symbolSize: [30, 100]  
      }  
    }  
  }  
]
```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    outOfRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in outOfRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus series.data) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von [visualMap.min, visualMap.max] in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei [visualMap.min, visualMap.max] auf [0, 100] gesetzt, und sei series.data [50, 10, 100]. Diese Werte sollen auf den opacity-Bereich [0.4, 1] gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die visualMap-Komponente berechnet dann linear die opacity-Werte [0.7, 0.44, 1] für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel opacity: [1, 0.4], wodurch das Endergebnis für die obige Serie [0.7, 0.96, 0.4] beträgt.

Hinweis: [visualMap.min, visualMap.max] sollte manuell gesetzt werden und ist standardmäßig [0, 200], nicht dataMin und dataMax aus series.data.

Wie muss die visualMap-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige visualMap wird verwendet, oder
- Eine stückweise visualMap wird verwendet und visualMap.categories ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. color: ['#333', '#777'].

- Einzelne Zahlen oder Strings können auch verwendet werden, die visualMap-Komponente wandelt solche in ein Array um. Zum Beispiel: opacity: 0.4 wird konvertiert zu opacity: [0.4, 0.4], color: '#333' wird konvertiert zu color: ['#333', '#333'].
- Für die visuellen Kanäle symbolSize, opacity, colorAlpha, colorLightness, colorSaturation und colorHue hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von visualMap.min, visueller Wert von visualMap.max]. Zum Beispiel: colorLightness: [0.8, 0.2] bedeutet, dass ein Datenwert in series.data der gleich visualMap.min ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich visualMap.max ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen [visualMap.min, visualMap.max] und [0.8, 0.2] gemappt.
- Für den visuellen Kanal color wird ein Array wie ['#333', '#78ab23', 'blue'] verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich visualMap.min sind, auf '#333' gemappt, Datenwerte, die gleich visualMap.max sind, auf 'blue' gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal symbol wird ein Array wie ['circle', 'rect', 'diamond'] verwendet, mit dem Datenwerte, die gleich visualMap.min sind, auf 'circle' gemappt werden, Datenwerte, die gleich visualMap.max sind, auf 'diamond' gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu visualMap.min und visualMap.max auf 'circle', 'rect' oder 'diamond' gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- opacity, colorAlpha, colorLightness, colorSaturation  
Die möglichen Werte liegen in [0, 1].
- colorHue  
Die möglichen Werte liegen in [0, 360].
- color  
Mögliche Werte sind RGB-Ausdrücke, wie 'rgb(128, 128, 128)', RGBA-Ausdrücke, wie 'rgba(128, 128, 128, 0.5)', oder Hex-Ausdrücke, wie '#ccc'.
- symbol  
Von ECharts bereitgestellte Icon-Typen sind 'circle', 'rect', 'roundRect', 'triangle', 'diamond', 'pin', 'arrow' und 'none'.  
Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.  
Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:



```

visualMap: {
  type: 'piecewise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  outOfRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
  Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
}

```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem `setOption` aufgerufen wurde um die initiale Option zu setzen), kann `setOption` erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```

chart.setOption({
  visualMap: {
    outOfRange: {color: ['red', 'blue']}
  }
});

```

#### Hinweis:

- Diese `visualMap`-Eigenschaften (`inRange`, `outOfRange`, `target`, `controller`) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn `setOption` erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an `setOption` übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird `getOption()` -> modifiziere die erhaltene Option -> `setOption(modifizierte Option)`, strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.outOfRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.outOfRange.color = ['red', 'blue'];
option.visualMap.controller.outOfRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

### 5.26.22 chart.visualMap.controller.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Eigenschaften `inRange` und `outOfRange` können im `controller`-Objekt gesetzt werden, wodurch diese Werte für für `inRange` und `outOfRange` nur vom Controller (der `visualMap`-Komponente selbst) und nicht vom Diagramm (der Serie) benutzt werden. Diese Eigenschaft ist nützlich in Szenarien, in denen die Ansicht des Controllers im Detail angepasst werden muss.

### 5.26.23 chart.visualMap.controller.inRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

inRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    inRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      inRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    inRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in inRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  inRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    inRange: {color: ['red', 'blue']}
  }
});
```

**Hinweis:**

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.inRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.inRange.color = ['red', 'blue'];
option.visualMap.controller.inRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

**Hinweis:** Es gibt Standardfarben ['#f6efa6', '#d88273', '#bf444c'] in inRange wenn inRange nicht gesetzt wird. Sind diese unerwünscht, können sie durch inRange: { color: null } deaktiviert werden.

### 5.26.24 chart.visualMap.controller.outOfRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

outOfRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    outOfRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:



```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      outOfRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    outOfRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in outOfRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  outOfRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    outOfRange: {color: ['red', 'blue']}
  }
});
```

## Hinweis:

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.outOfRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.outOfRange.color = ['red', 'blue'];
option.visualMap.controller.outOfRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

## 5.26.25 chart.visualMap.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 5.26.26 chart.visualMap.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 4 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 5.26.27 chart.visualMap.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Abstand zwischen der visualMap-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 5.26.28 chart.visualMap.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der visualMap-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom
- auto

### 5.26.29 chart.visualMap.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Abstand zwischen der visualMap-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

### Erlaubte Werte

- {percent}
- {integer}
- auto

## 5.26.30 chart.visualMap.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Abstand zwischen der visualMap-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

### Erlaubte Werte

- {integer}
- {percent}

## 5.26.31 chart.visualMap.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'vertical' ]

Wie das Layout der visualMap-Komponente aussehen soll.

#### Erlaubte Werte

- horizontal
- vertical

### 5.26.32 chart.visualMap.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der visualMap um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 5.26.33 chart.visualMap.backgroundColor

<b>Datentyp</b>	COLOR
-----------------	-------



<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(0,0,0,0)' ]

Hintergrundfarbe der visualMap-Komponente

### 5.26.34 chart.visualMap.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe der visualMap-Komponente.

### 5.26.35 chart.visualMap.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite der visualMap-Komponente, mit Einheit px.

### 5.26.36 chart.visualMap.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#bf444c', '#d88273', '#f6efa6'] ]

Diese Eigenschaft verbleibt nur für Kompatibilität mit ECharts 2 und ihre Verwendung in ECharts 3 wird nicht empfohlen. Farben sollten stattdessen in `visualMap.inRange` oder `visualMap.outOfRange` konfiguriert werden.

Wird die Eigenschaft weiter verwendet, sollte das folgende Problem bemerkt werden: Die Reihenfolge, in der die Datenwerte auf Farbwerte aus der Eigenschaft `color` gemappt werden, ist von groß nach klein, während sie für die Eigenschaften `visualMap.inRange` und `visualMap.outOfRange` von klein nach groß ist.

### 5.26.37 chart.visualMap.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Textes der `visualMap`.

### 5.26.38 chart.visualMap.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: #333 ]

Textfarbe.

## 5.26.39 chart.visualMap.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 5.26.40 chart.visualMap.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'

- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 5.26.41 chart.visualMap.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

visualMap Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 5.26.42 chart.visualMap.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 5.26.43 chart.visualMap.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 5.26.44 chart.visualMap.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes

Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.26.45 chart.visualMap.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 5.26.46 chart.visualMap.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 5.26.47 chart.visualMap.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 5.26.48 chart.visualMap.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 5.26.49 chart.visualMap.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 5.26.50 chart.visualMap.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 5.26.51 chart.visualMap.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 5.26.52 chart.visualMap.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Formatierer für die Beschriftung.

- Wird er als String gesetzt, steht er für eine String-Template, zum Beispiel : 'aaaa{value}bbbb', wobei {value} den Wert am Rand des ausgewählten Bereichs darstellt.
- Wird er als Funktion gesetzt, steht er für eine Callback-Funktion, zum Beispiel:



```
formatter: function (value) {
  return 'aaaa' + value + 'bbbb';
}
```

### 5.26.53 chart.visualMap.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Stetige Daten können nach splitNumber in Stücke unterteilt werden. Das heißt, wenn splitNumber 5 ist, werden die Daten in 5 Stücke unterteilt.

Der Bereich der stetigen Daten sollte mit max und min definiert werden.

Wenn visualMap.pieces oder visualMap.categories gesetzt ist, wird splitNumber nicht mehr verwendet.

### 5.26.54 chart.visualMap.pieces

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wird verwendet um anzupassen, wie stetige Daten aufgeteilt werden sollen, und um spezifische Ansichten für solche Stücke zu definieren. Zum Beispiel:

```
pieces: [
  // Der Bereich eines Stücks kann über die Eigenschaften min und max definiert
  werden,
  // wobei min auf -Infinity gesetzt wird, falls ignoriert,
  // und max auf Infinity gesetzt wird, falls ignoriert.
  {min: 1500},
  {min: 900, max: 1500},
  {min: 310, max: 1000},
  {min: 200, max: 300},
  // Die Beschriftung eines Stücks kann spezifiziert werden.
  {min: 10, max: 200, label: '10 to 200 (custom label) '},
  // Die Farbe eines Stücks kann spezifiziert werden.
  {value: 123, label: '123 (custom special color) ', color: 'grey'},
  {max: 5}
]
```

Die folgenden visuellen Kanäle können für jedes Stück angepasst werden:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbton in HSL.

Hinweis: In ECharts2 heißt pieces splitList, was in ECharts3 für Kompatibilität erhalten bleibt. Es wird aber empfohlen, pieces zu verwenden.

Die Sequenz der Stücke lässt sich über einfaches ausprobieren herausfinden. Für weitere Informationen siehe visualMap.inverse.

### 5.26.55 chart.visualMap.categories

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn die Datenwerte in series.data (spezifiziert durch visualMap.dimension) diskret sind (auch bekannt als Kategoriedaten oder abzählbare Daten), und ein **Table Mapping** von den Datenwerten zu visuellen Kanälen vorgenommen werden soll, wird categories verwendet, um die gesamte Aufzählung der Daten zu beschreiben. Zum Beispiel:

```
categories: [
  'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
],
```

Die Reihenfolge in categories kann durch ausprobieren erkannt werden. Für weitere Informationen siehe `visualMap.inverse`.

### 5.26.56 chart.visualMap.minOpen

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option funktioniert wenn type auf 'piecewise' steht und min/max/splitNumber gesetzt sind. Steht dieser Wert auf true, so wird ein zusätzliches Stück mit der Beschriftung "< min" angezeigt.

### 5.26.57 chart.visualMap.maxOpen

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option funktioniert wenn type auf 'piecewise' steht und min/max/splitNumber gesetzt sind. Steht dieser Wert auf true, so wird ein zusätzliches Stück mit der Beschriftung "> max" angezeigt.

### 5.26.58 chart.visualMap.selectedMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'multiple' ]

Wechselt zwischen Einzelauswahl und multipler Auswahl.

Erlaubte Werte

- single
- multiple

### 5.26.59 chart.visualMap.showLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Beschriftung jedes Elements angezeigt werden soll. Standardmäßig wird die Beschriftung nicht angezeigt, wenn visualMap.text verwendet wird, sonst wird sie angezeigt.

## 6 ECharts (5) [Options]

<b>Name</b>	ECharts
<b>Version</b>	5
<b>Hersteller</b>	Apache
<b>Webseite</b>	<a href="https://ecomfe.github.io/echarts-doc/public/en/">https://ecomfe.github.io/echarts-doc/public/en/</a>

### 6.1 ECharts (5) Angle Axis

#### 6.1.1 chart.angleAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Winkelachse im Polarkoordinatensystem.

#### 6.1.2 chart.angleAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.1.3 chart.angleAxis.polarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index der Winkelachse im Polarkoordinatensystem. Standardmäßig wird die erste Achse verwendet.

### 6.1.4 chart.angleAxis.startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 90 ]

Startwinkel der Achse. Standardmäßig 90 Grad, was für von der Mitte aus nach oben zeigend steht. 0 Grad steht für von der Mitte aus nach rechts zeigend.

### 6.1.5 chart.angleAxis.clockwise

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Werte der Winkelachse im Uhrzeigersinn oder gegen den Uhrzeigersinn ansteigen. Standardmäßig true für im Uhrzeigersinn.

### 6.1.6 chart.angleAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'category' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 6.1.7 chart.angleAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

## 6.1.8 chart.angleAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 6.1.9 chart.angleAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.1.10 chart.angleAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 6.1.11 chart.angleAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.1.12 chart.angleAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.1.13 chart.angleAxis.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den

meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.1.14 chart.angleAxis.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.1.15 chart.angleAxis.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 6.1.16 chart.angleAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```

{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}

```

### 6.1.17 chart.angleAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 6.1.18 chart.angleAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

## 6.1.19 chart.angleAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

### Erlaubte Werte

- none
- arrow

## 6.1.20 chart.angleAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.1.21 chart.angleAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.1.22 chart.angleAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.1.23 chart.angleAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

## Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.1.24 chart.angleAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.1.25 chart.angleAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.26 chart.angleAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.27 chart.angleAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.28 chart.angleAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.29 chart.angleAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.30 chart.angleAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.31 chart.angleAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.1.32 chart.angleAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.1.33 chart.angleAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.1.34 chart.angleAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.1.35 chart.angleAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.1.36 chart.angleAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.1.37 chart.angleAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Linienstil der Achsenticks.

### 6.1.38 chart.angleAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Linien der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.1.39 chart.angleAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.1.40 chart.angleAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.41 chart.angleAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.42 chart.angleAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.43 chart.angleAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.44 chart.angleAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.45 chart.angleAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.46 chart.angleAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 6.1.47 chart.angleAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.1.48 chart.angleAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.1.49 chart.angleAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.1.50 chart.angleAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 6.1.51 chart.angleAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 6.1.52 chart.angleAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.1.53 chart.angleAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.1.54 chart.angleAxis.axisLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {  
  color: function (value, index) {  
    return value >= 0 ? 'green' : 'red';  
  }  
}
```

### 6.1.55 chart.angleAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.1.56 chart.angleAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.1.57 chart.angleAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.1.58 chart.angleAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.1.59 chart.angleAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.1.60 chart.angleAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 6.1.61 chart.angleAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.1.62 chart.angleAxis.axisLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.1.63 chart.angleAxis.axisLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.1.64 chart.angleAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.1.65 chart.angleAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.1.66 chart.angleAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.1.67 chart.angleAxis.axisLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.1.68 chart.angleAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.1.69 chart.angleAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.1.70 chart.angleAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.1.71 chart.angleAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.1.72 chart.angleAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.1.73 chart.angleAxis.axisLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.1.74 chart.angleAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.1.75 chart.angleAxis.axisLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.1.76 chart.angleAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.1.77 chart.angleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.1.78 chart.angleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.1.79 chart.angleAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.1.80 chart.angleAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.1.81 chart.angleAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.1.82 chart.angleAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.1.83 chart.angleAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.



## 6.1.84 chart.angleAxis.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc'] ]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

## 6.1.85 chart.angleAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.1.86 chart.angleAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.87 chart.angleAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.88 chart.angleAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.89 chart.angleAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.90 chart.angleAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.91 chart.angleAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.92 chart.angleAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.1.93 chart.angleAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.1.94 chart.angleAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.1.95 chart.angleAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 6.1.96 chart.angleAxis.splitArea.areaStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.1.97 chart.angleAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.98 chart.angleAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.99 chart.angleAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.100 chart.angleAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.101 chart.angleAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.102 chart.angleAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 6.1.103 chart.angleAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 6.1.104 chart.angleAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.1.105 chart.angleAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.1.106 chart.angleAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.1.107 chart.angleAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.1.108 chart.angleAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.1.109 chart.angleAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

## 6.1.110 chart.angleAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

#### Parameter:

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 6.1.111 chart.angleAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 6.1.112 chart.angleAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.1.113 chart.angleAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 6.1.114 chart.angleAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.1.115 chart.angleAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.1.116 chart.angleAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.1.117 chart.angleAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.1.118 chart.angleAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.1.119 chart.angleAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.1.120 chart.angleAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
-----------------	-------



<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.1.121 chart.angleAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.1.122 chart.angleAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.1.123 chart.angleAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.1.124 chart.angleAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.1.125 chart.angleAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.1.126 chart.angleAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.1.127 chart.angleAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.1.128 chart.angleAxis.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.1.129 chart.angleAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.1.130 chart.angleAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.131 chart.angleAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.132 chart.angleAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.133 chart.angleAxis.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.134 chart.angleAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 6.1.135 chart.angleAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

### 6.1.136 chart.angleAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.1.137 chart.angleAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.138 chart.angleAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.139 chart.angleAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.140 chart.angleAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.141 chart.angleAxis.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.142 chart.angleAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.143 chart.angleAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.1.144 chart.angleAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.1.145 chart.angleAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.146 chart.angleAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.147 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.148 chart.angleAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.149 chart.angleAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.150 chart.angleAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.1.151 chart.angleAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.1.152 chart.angleAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 6.1.153 chart.angleAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.1.154 chart.angleAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 6.1.155 chart.angleAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über



das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.1.156 chart.angleAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.1.157 chart.angleAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.1.158 chart.angleAxis.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.1.159 chart.angleAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.1.160 chart.angleAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.161 chart.angleAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.162 chart.angleAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.163 chart.angleAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.164 chart.angleAxis.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.1.165 chart.angleAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.1.166 chart.angleAxis.axisLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.1.167 chart.angleAxis.axisLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.168 chart.angleAxis.axisLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.1.169 chart.angleAxis.axisLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.170 chart.angleAxis.axisTick.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.1.171 chart.angleAxis.axisTick.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.172 chart.angleAxis.axisTick.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.1.173 chart.angleAxis.axisTick.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.



Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.174 chart.angleAxis.axisLabel.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.1.175 chart.angleAxis.axisLabel.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.1.176 chart.angleAxis.axisLabel.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate
- break
- breakAll

### 6.1.177 chart.angleAxis.axisLabel.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.1.178 chart.angleAxis.axisLabel.textBorderStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

Erlaubte Werte

- dashed
- dotted
- solid

### 6.1.179 chart.angleAxis.axisLabel.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderType` können wir den Linienstil flexibler gestalten.

### 6.1.180 chart.angleAxis.splitLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `Type` können wir den Linienstil flexibler gestalten.

### 6.1.181 chart.angleAxis.splitLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.182 chart.angleAxis.splitLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.

- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

### Erlaubte Werte

- bevel
- round
- miter

### 6.1.183 chart.angleAxis.splitLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.184 chart.angleAxis.axisPointer.label.textBorderStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.1.185 chart.angleAxis.axisPointer.label.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderType` können wir den Linienstil flexibler gestalten.

### 6.1.186 chart.angleAxis.axisLabel.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.1.187 chart.angleAxis.axisLabel.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.1.188 chart.angleAxis.axisPointer.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.1.189 chart.angleAxis.axisPointer.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.190 chart.angleAxis.axisPointer.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter



### 6.1.191 chart.angleAxis.axisPointer.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.192 chart.angleAxis.minorSplitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Kleinere Teilungslinien der Achse im Rasterbereich. Es wird am minorTick ausgerichtet

### 6.1.193 chart.angleAxis.minorSplitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Minor-Split-Lines angezeigt werden soll.

### 6.1.194 chart.angleAxis.minorSplitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des MinorSplitLine -Linienstils

### 6.1.195 chart.angleAxis.minorSplitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

minorSplitLineLine Farbe

### 6.1.196 chart.angleAxis.minorSplitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der minorSplitLines.

### 6.1.197 chart.angleAxis.minorSplitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Zeilentyp. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit dashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ type: [5, 10], dashOffset: 5 }
```

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.198 chart.angleAxis.minorSplitLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.1.199 chart.angleAxis.minorSplitLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.200 chart.angleAxis.minorSplitLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.

- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.1.201 chart.angleAxis.minorSplitLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.202 chart.angleAxis.minorSplitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.203 chart.angleAxis.minorSplitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.204 chart.angleAxis.minorSplitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.205 chart.angleAxis.minorSplitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.206 chart.angleAxis.minorSplitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.207 chart.angleAxis.minorTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Einstellungen im Zusammenhang mit kleinen Häkchen. Hinweis: minorTick ist in der Kategorietyppachse nicht verfügbar.

### 6.1.208 chart.angleAxis.minorTick.show

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die MinorAxis angezeigt werden soll.

### 6.1.209 chart.angleAxis.minorTick.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl des Intervalls geteilt durch kleine Ticks.

### 6.1.210 chart.angleAxis.minorTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Länge der kleinen Tick-Linien.



### 6.1.211 chart.angleAxis.minorTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des MinorTick-Linienstils

### 6.1.212 chart.angleAxis.minorTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stilkonfiguration kleiner Teilstriche

### 6.1.213 chart.angleAxis.minorTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der MinorTicks.

### 6.1.214 chart.angleAxis.minorTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Zeilentyp. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit dashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ type: [5, 10], dashOffset: 5 }
```

Erlaubte Werte

- solid
- dashed
- dotted

### 6.1.215 chart.angleAxis.minorTick.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.1.216 chart.angleAxis.minorTick.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.1.217 chart.angleAxis.minorTick.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.

- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.1.218 chart.angleAxis.minorTick.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.1.219 chart.angleAxis.minorTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.1.220 chart.angleAxis.minorTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.1.221 chart.angleAxis.minorTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.1.222 chart.angleAxis.minorTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.1.223 chart.angleAxis.minorTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.1.224 chart.angleAxis.maxInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Maximaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.1.225 chart.axisPointer.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Angle Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

## 6.2 ECharts (5) Axis

### 6.2.1 chart.yAxis.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Y-Achse in kartesischen (rechteckigen) Koordinatensystemen. Normalerweise kann eine einzelne Gitterkomponente höchstens 2 Y-Achsen platzieren, eine links und eine rechts. Ein Offset kann verwendet werden, um Überlappung bei mehr als zwei Y-Achsen zu vermeiden.

### 6.2.2 chart.yAxis.{x}.id

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.2.3 chart.yAxis.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Y-Achse angezeigt werden soll.

### 6.2.4 chart.yAxis.{x}.gridIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index des Gitters zu dem die Y-Achse gehört. Standardmäßig wird das erste Gitter genommen.



## 6.2.5 chart.yAxis.{x}.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position der Y-Achse.

Die erste Y-Achse im Gitter wird standardmäßig auf der linken Seite ('left') platziert, und die zweite auf der gegenüberliegenden Seite zur ersten Y-Achse.

Erlaubte Werte

- left
- right

## 6.2.6 chart.yAxis.{x}.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset der Y-Achse relativ zur Standardposition. Nützlich wenn mehrere Y-Achsen den gleichen Positionswert haben.

## 6.2.7 chart.yAxis.{x}.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

## 6.2.8 chart.yAxis.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

## 6.2.9 chart.yAxis.{x}.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 6.2.10 chart.yAxis.{x}.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 6.2.11 chart.yAxis.{x}.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

## 6.2.12 chart.yAxis.{x}.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.2.13 chart.yAxis.{x}.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold

- bolder
- lighter
- {integer}

### 6.2.14 chart.yAxis.{x}.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.2.15 chart.yAxis.{x}.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Axis-Name Schriftgröße.

### 6.2.16 chart.yAxis.{x}.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.2.17 chart.yAxis.{x}.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.2.18 chart.yAxis.{x}.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.2.19 chart.yAxis.{x}.nameTextStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.2.20 chart.yAxis.{x}.nameTextStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.2.21 chart.yAxis.{x}.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.2.22 chart.yAxis.{x}.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.2.23 chart.yAxis.{x}.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.2.24 chart.yAxis.{x}.nameTextStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.2.25 chart.yAxis.{x}.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.2.26 chart.yAxis.{x}.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.2.27 chart.yAxis.{x}.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

## 6.2.28 chart.yAxis.{x}.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.29 chart.yAxis.{x}.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.30 chart.yAxis.{x}.nameTextStyle.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.2.31 chart.yAxis.{x}.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.2.32 chart.yAxis.{x}.nameTextStyle.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.33 chart.yAxis.{x}.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.34 chart.yAxis.{x}.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.35 chart.yAxis.{x}.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.2.36 chart.yAxis.{x}.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.2.37 chart.yAxis.{x}.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Lücke zwischen Achsenname und Achsenlinie.

### 6.2.38 chart.yAxis.{x}.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 6.2.39 chart.yAxis.{x}.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist.

### 6.2.40 chart.yAxis.{x}.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 6.2.41 chart.yAxis.{x}.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.2.42 chart.yAxis.{x}.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 6.2.43 chart.yAxis.{x}.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

## 6.2.44 chart.yAxis.{x}.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.2.45 chart.yAxis.{x}.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.2.46 chart.yAxis.{x}.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.2.47 chart.yAxis.{x}.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.2.48 chart.yAxis.{x}.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 6.2.49 chart.yAxis.{x}.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 6.2.50 chart.yAxis.{x}.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

## 6.2.51 chart.yAxis.{x}.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.2.52 chart.yAxis.{x}.axisLine.onZero

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X- oder Y-Achse auf dem Ursprung der jeweils anderen liegt, wo der Achsenwert 0 ist. Nur gültig, wenn die andere Achse den Typ 'value' hat und den Wert 0 enthält.

### 6.2.53 chart.yAxis.{x}.axisLine.onZeroAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere Achsen existieren, kann dieser Wert verwendet werden, um die Achse zu spezifizieren, auf die sich 'onZero' bezieht.

### 6.2.54 chart.yAxis.{x}.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.2.55 chart.yAxis.{x}.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.2.56 chart.yAxis.{x}.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.2.57 chart.yAxis.{x}.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.2.58 chart.yAxis.{x}.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 6.2.59 chart.yAxis.{x}.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 6.2.60 chart.yAxis.{x}.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 6.2.61 chart.yAxis.{x}.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.62 chart.yAxis.{x}.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.63 chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.64 chart.yAxis.{x}.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.65 chart.yAxis.{x}.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.66 chart.yAxis.{x}.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.2.67 chart.yAxis.{x}.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.2.68 chart.yAxis.{x}.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.2.69 chart.yAxis.{x}.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.70 chart.yAxis.{x}.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.2.71 chart.yAxis.{x}.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.2.72 chart.yAxis.{x}.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.2.73 chart.yAxis.{x}.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.2.74 chart.yAxis.{x}.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.2.75 chart.yAxis.{x}.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

### Erlaubte Werte

- solid
- dashed
- dotted

## 6.2.76 chart.yAxis.{x}.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.2.77 chart.yAxis.{x}.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 6.2.78 chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.79 chart.yAxis.{x}.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.80 chart.yAxis.{x}.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.81 chart.yAxis.{x}.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenbeschriftung gehören.

### 6.2.82 chart.yAxis.{x}.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.2.83 chart.yAxis.{x}.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

## 6.2.84 chart.yAxis.{x}.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

## 6.2.85 chart.yAxis.{x}.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

## 6.2.86 chart.yAxis.{x}.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

## 6.2.87 chart.yAxis.{x}.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

## 6.2.88 chart.yAxis.{x}.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.2.89 chart.yAxis.{x}.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.2.90 chart.yAxis.{x}.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.2.91 chart.yAxis.{x}.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.2.92 chart.yAxis.{x}.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.2.93 chart.yAxis.{x}.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.2.94 chart.yAxis.{x}.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.2.95 chart.yAxis.{x}.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.2.96 chart.yAxis.{x}.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.2.97 chart.yAxis.{x}.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.2.98 chart.yAxis.{x}.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

## 6.2.99 chart.yAxis.{x}.axisLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.2.100 chart.yAxis.{x}.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.2.101 chart.yAxis.{x}.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.2.102 chart.yAxis.{x}.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.2.103 chart.yAxis.{x}.axisLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.2.104 chart.yAxis.{x}.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.2.105 chart.yAxis.{x}.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.2.106 chart.yAxis.{x}.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.2.107 chart.yAxis.{x}.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.108 chart.yAxis.{x}.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.109 chart.yAxis.{x}.axisLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.2.110 chart.yAxis.{x}.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.2.111 chart.yAxis.{x}.axisLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.112 chart.yAxis.{x}.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.113 chart.yAxis.{x}.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.114 chart.yAxis.{x}.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.2.115 chart.yAxis.{x}.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.2.116 chart.yAxis.{x}.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.2.117 chart.yAxis.{x}.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.2.118 chart.yAxis.{x}.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.119 chart.yAxis.{x}.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 6.2.120 chart.yAxis.{x}.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 6.2.121 chart.yAxis.{x}.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.2.122 chart.yAxis.{x}.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.2.123 chart.yAxis.{x}.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.124 chart.yAxis.{x}.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.125 chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.126 chart.yAxis.{x}.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.127 chart.yAxis.{x}.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.128 chart.yAxis.{x}.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.2.129 chart.yAxis.{x}.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.130 chart.yAxis.{x}.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.2.131 chart.yAxis.{x}.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 6.2.132 chart.yAxis.{x}.splitArea.areaStyle.color

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.2.133 chart.yAxis.{x}.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.134 chart.yAxis.{x}.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.135 chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.136 chart.yAxis.{x}.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.137 chart.yAxis.{x}.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.138 chart.yAxis.{x}.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 6.2.139 chart.yAxis.{x}.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 6.2.140 chart.yAxis.{x}.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.2.141 chart.yAxis.{x}.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.2.142 chart.yAxis.{x}.axisPointer.z

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.2.143 chart.yAxis.{x}.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.2.144 chart.yAxis.{x}.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.2.145 chart.yAxis.{x}.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 6.2.146 chart.yAxis.{x}.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.2.147 chart.yAxis.{x}.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 6.2.148 chart.yAxis.{x}.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.2.149 chart.yAxis.{x}.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 6.2.150 chart.yAxis.{x}.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.2.151 chart.yAxis.{x}.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.2.152 chart.yAxis.{x}.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.2.153 chart.yAxis.{x}.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.2.154 chart.yAxis.{x}.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.155 chart.yAxis.{x}.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.156 chart.yAxis.{x}.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.2.157 chart.yAxis.{x}.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.2.158 chart.yAxis.{x}.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.159 chart.yAxis.{x}.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.160 chart.yAxis.{x}.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.161 chart.yAxis.{x}.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.2.162 chart.yAxis.{x}.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.2.163 chart.yAxis.{x}.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.2.164 chart.yAxis.{x}.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.2.165 chart.yAxis.{x}.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.2.166 chart.yAxis.{x}.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.167 chart.yAxis.{x}.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.168 chart.yAxis.{x}.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.169 chart.yAxis.{x}.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.170 chart.yAxis.{x}.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 6.2.171 chart.yAxis.{x}.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.2.172 chart.yAxis.{x}.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.2.173 chart.yAxis.{x}.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.2.174 chart.yAxis.{x}.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.175 chart.yAxis.{x}.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.176 chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.177 chart.yAxis.{x}.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.178 chart.yAxis.{x}.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.179 chart.yAxis.{x}.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.2.180 chart.yAxis.{x}.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.2.181 chart.yAxis.{x}.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.182 chart.yAxis.{x}.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.183 chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.184 chart.yAxis.{x}.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.185 chart.yAxis.{x}.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.186 chart.yAxis.{x}.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.2.187 chart.yAxis.{x}.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.2.188 chart.yAxis.{x}.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status.

Erlaubte Werte

- show
- hide

### 6.2.189 chart.yAxis.{x}.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.2.190 chart.yAxis.{x}.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 6.2.191 chart.yAxis.{x}.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.2.192 chart.yAxis.{x}.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.2.193 chart.yAxis.{x}.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.2.194 chart.yAxis.{x}.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.2.195 chart.yAxis.{x}.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.2.196 chart.yAxis.{x}.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.197 chart.yAxis.{x}.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.198 chart.yAxis.{x}.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.199 chart.yAxis.{x}.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.200 chart.yAxis.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente der Y-Achse.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.2.201 chart.yAxis.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente der Y-Achse, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.2.202 chart.xAxis.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die X-Achse in kartesischen (rechteckigen) Koordinatensystemen. Normalerweise kann eine einzelne Gitterkomponente höchstens 2 X-Achsen platzieren, eine links und eine rechts. Ein Offset kann verwendet werden, um Überlappung bei mehr als zwei X-Achsen zu vermeiden.

### 6.2.203 chart.xAxis.{x}.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.2.204 chart.xAxis.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X-Achse angezeigt werden soll.

### 6.2.205 chart.xAxis.{x}.gridIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index des Gitters zu dem die X-Achse gehört. Standardmäßig wird das erste Gitter genommen.

### 6.2.206 chart.xAxis.{x}.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position der X-Achse.

Die erste X-Achse im Gitter wird standardmäßig unten platziert, und die zweite auf der gegenüberliegenden Seite zur ersten X-Achse.

#### Erlaubte Werte

- top
- bottom



### 6.2.207 chart.xAxis.{x}.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset der X-Achse relativ zur Standardposition. Nützlich wenn mehrere X-Achsen den gleichen Positionswert haben.

### 6.2.208 chart.xAxis.{x}.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheiden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 6.2.209 chart.xAxis.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

### 6.2.210 chart.xAxis.{x}.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 6.2.211 chart.xAxis.{x}.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 6.2.212 chart.xAxis.{x}.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 6.2.213 chart.xAxis.{x}.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.2.214 chart.xAxis.{x}.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.2.215 chart.xAxis.{x}.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie des Achsennamens.

Akzeptiert auch 'serif', 'monospace', ...

### 6.2.216 chart.xAxis.{x}.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 6.2.217 chart.xAxis.{x}.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.2.218 chart.xAxis.{x}.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.2.219 chart.xAxis.{x}.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.2.220 chart.xAxis.{x}.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.2.221 chart.xAxis.{x}.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.2.222 chart.xAxis.{x}.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.2.223 chart.xAxis.{x}.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.



### 6.2.224 chart.xAxis.{x}.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.2.225 chart.xAxis.{x}.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.2.226 chart.xAxis.{x}.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.2.227 chart.xAxis.{x}.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.2.228 chart.xAxis.{x}.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.2.229 chart.xAxis.{x}.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.230 chart.xAxis.{x}.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.231 chart.xAxis.{x}.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.2.232 chart.xAxis.{x}.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.2.233 chart.xAxis.{x}.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.234 chart.xAxis.{x}.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.235 chart.xAxis.{x}.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.236 chart.xAxis.{x}.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.2.237 chart.xAxis.{x}.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.2.238 chart.xAxis.{x}.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 6.2.239 chart.xAxis.{x}.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 6.2.240 chart.xAxis.{x}.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 6.2.241 chart.xAxis.{x}.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 6.2.242 chart.xAxis.{x}.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.



Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.2.243 chart.xAxis.{x}.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.2.244 chart.xAxis.{x}.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 6.2.245 chart.xAxis.{x}.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.2.246 chart.xAxis.{x}.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.2.247 chart.xAxis.{x}.interval

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.2.248 `chart.xAxis.{x}.logBase`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.2.249 `chart.xAxis.{x}.silent`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 6.2.250 chart.xAxis.{x}.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 6.2.251 chart.xAxis.{x}.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Einstellungen, die zu axisLine gehören.

### 6.2.252 chart.xAxis.{x}.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.2.253 chart.xAxis.{x}.axisLine.onZero

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die X- oder Y-Achse auf dem Ursprung der jeweils anderen liegt, wo der Achsenwert 0 ist. Nur gültig, wenn die andere Achse den Typ 'value' hat und den Wert 0 enthält.

### 6.2.254 chart.xAxis.{x}.axisLine.onZeroAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere Achsen existieren, kann dieser Wert verwendet werden, um die Achse zu spezifizieren, auf die sich 'onZero' bezieht.

### 6.2.255 chart.xAxis.{x}.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.2.256 chart.xAxis.{x}.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.2.257 chart.xAxis.{x}.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.2.258 chart.xAxis.{x}.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.2.259 chart.xAxis.{x}.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```



### 6.2.260 chart.xAxis.{x}.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.2.261 chart.xAxis.{x}.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

### 6.2.262 chart.xAxis.{x}.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.263 chart.xAxis.{x}.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.264 chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.265 chart.xAxis.{x}.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.266 chart.xAxis.{x}.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.267 chart.xAxis.{x}.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Einstellungen, die zu den Achsenticks gehören.

### 6.2.268 chart.xAxis.{x}.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.2.269 chart.xAxis.{x}.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.2.270 chart.xAxis.{x}.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.271 chart.xAxis.{x}.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.2.272 chart.xAxis.{x}.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.2.273 chart.xAxis.{x}.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.2.274 chart.xAxis.{x}.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.2.275 chart.xAxis.{x}.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.2.276 chart.xAxis.{x}.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.2.277 chart.xAxis.{x}.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.278 chart.xAxis.{x}.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.279 chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.280 chart.xAxis.{x}.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.281 chart.xAxis.{x}.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
-----------------	---------



<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.282 chart.xAxis.{x}.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 6.2.283 chart.xAxis.{x}.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.2.284 chart.xAxis.{x}.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.285 chart.xAxis.{x}.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.2.286 chart.xAxis.{x}.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

Beispiel:

```
dashlet.addOption("chart.xAxis.axisLabel.rotate", "45");
```

Siehe auch :

- chart.grid.height
- chart.grid.bottom

## 6.2.287 chart.xAxis.{x}.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

## 6.2.288 chart.xAxis.{x}.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```

// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}

```

### 6.2.289 chart.xAxis.{x}.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.2.290 chart.xAxis.{x}.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.2.291 chart.xAxis.{x}.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig axisLine.lineStyle.color. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.2.292 chart.xAxis.{x}.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.2.293 chart.xAxis.{x}.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.2.294 chart.xAxis.{x}.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.2.295 chart.xAxis.{x}.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.2.296 chart.xAxis.{x}.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.2.297 chart.xAxis.{x}.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom



## 6.2.298 chart.xAxis.{x}.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.2.299 chart.xAxis.{x}.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.2.300 chart.xAxis.{x}.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.2.301 chart.xAxis.{x}.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.2.302 chart.xAxis.{x}.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.2.303 chart.xAxis.{x}.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.2.304 chart.xAxis.{x}.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.2.305 chart.xAxis.{x}.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.2.306 chart.xAxis.{x}.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.2.307 chart.xAxis.{x}.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.2.308 chart.xAxis.{x}.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.309 chart.xAxis.{x}.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.2.310 chart.xAxis.{x}.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.2.311 chart.xAxis.{x}.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.2.312 chart.xAxis.{x}.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.313 chart.xAxis.{x}.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.314 chart.xAxis.{x}.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.315 chart.xAxis.{x}.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.2.316 chart.xAxis.{x}.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.2.317 chart.xAxis.{x}.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.2.318 chart.xAxis.{x}.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.2.319 chart.xAxis.{x}.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.320 chart.xAxis.{x}.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

## 6.2.321 chart.xAxis.{x}.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc'] ]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbararray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {  
  lineStyle: {  
    // Dunkle und helle Farben werden abwechselnd verwendet  
    color: ['#aaa', '#ddd']  
  }  
}
```

## 6.2.322 chart.xAxis.{x}.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.2.323 chart.xAxis.{x}.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.2.324 chart.xAxis.{x}.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.325 chart.xAxis.{x}.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.326 chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.327 chart.xAxis.{x}.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.328 chart.xAxis.{x}.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.329 chart.xAxis.{x}.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.2.330 chart.xAxis.{x}.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.2.331 chart.xAxis.{x}.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.2.332 chart.xAxis.{x}.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 6.2.333 chart.xAxis.{x}.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)','rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.2.334 chart.xAxis.{x}.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.335 chart.xAxis.{x}.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



## 6.2.336 chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.2.337 chart.xAxis.{x}.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 6.2.338 chart.xAxis.{x}.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.339 chart.xAxis.{x}.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 6.2.340 chart.xAxis.{x}.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 6.2.341 chart.xAxis.{x}.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.2.342 chart.xAxis.{x}.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.2.343 chart.xAxis.{x}.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.2.344 chart.xAxis.{x}.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.2.345 chart.xAxis.{x}.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.2.346 chart.xAxis.{x}.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

## 6.2.347 chart.xAxis.{x}.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

### Parameter:

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.2.348 chart.xAxis.{x}.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 6.2.349 chart.xAxis.{x}.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.2.350 chart.xAxis.{x}.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 6.2.351 chart.xAxis.{x}.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.2.352 chart.xAxis.{x}.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.2.353 chart.xAxis.{x}.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.2.354 chart.xAxis.{x}.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.2.355 chart.xAxis.{x}.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.356 chart.xAxis.{x}.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.2.357 chart.xAxis.{x}.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.2.358 chart.xAxis.{x}.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.2.359 chart.xAxis.{x}.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.2.360 chart.xAxis.{x}.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.2.361 chart.xAxis.{x}.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.2.362 chart.xAxis.{x}.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.2.363 chart.xAxis.{x}.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.2.364 chart.xAxis.{x}.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.2.365 chart.xAxis.{x}.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.2.366 chart.xAxis.{x}.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.2.367 chart.xAxis.{x}.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.368 chart.xAxis.{x}.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.369 chart.xAxis.{x}.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.370 chart.xAxis.{x}.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.371 chart.xAxis.{x}.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 6.2.372 chart.xAxis.{x}.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.2.373 chart.xAxis.{x}.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.2.374 chart.xAxis.{x}.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.2.375 chart.xAxis.{x}.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.376 chart.xAxis.{x}.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.377 chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.378 chart.xAxis.{x}.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.379 chart.xAxis.{x}.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.380 chart.xAxis.{x}.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.2.381 chart.xAxis.{x}.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 6.2.382 chart.xAxis.{x}.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.383 chart.xAxis.{x}.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.384 chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.385 chart.xAxis.{x}.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.386 chart.xAxis.{x}.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.2.387 chart.xAxis.{x}.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.2.388 chart.xAxis.{x}.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.2.389 chart.xAxis.{x}.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status.

Erlaubte Werte

- show
- hide

### 6.2.390 chart.xAxis.{x}.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.2.391 chart.xAxis.{x}.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 6.2.392 chart.xAxis.{x}.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.2.393 chart.xAxis.{x}.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.2.394 chart.xAxis.{x}.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.2.395 chart.xAxis.{x}.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.2.396 chart.xAxis.{x}.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.2.397 chart.xAxis.{x}.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.2.398 chart.xAxis.{x}.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.2.399 chart.xAxis.{x}.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.2.400 chart.xAxis.{x}.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.2.401 chart.xAxis.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente der X-Achse.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.2.402 chart.xAxis.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

z-Wert aller grafischen Elemente der X-Achse, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

## 6.2.403 chart.axis.flip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Die Diagrammachse kann vertauscht werden.

Beispiel:

```
dashlet.addOption("chart.axis.flip", "false");
```

## 6.3 ECharts (5) Axis Pointer

### 6.3.1 chart.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dies ist die globale Konfigurierung von axisPointer.

axisPointer ist ein Werkzeug für die Darstellung von Referenzlinien und Achsenwerten unter dem Mauszeiger.

axisPointer kann auch auf Geräten mit Berührungssteuerung verwendet werden, wo Nutzer einen Knopf ziehen können, um Referenzlinien und Achsenwerte zu bewegen.

In Fällen in denen mehr als eine Achse existiert, hilft `axisPointer`, in die Daten zu sehen.

**Hinweis:** Im Allgemeinen wird der `axisPointer` für jede Achse die ihn braucht konfiguriert (zum Beispiel in `xAxis.axisPointer`) oder im Tooltip konfiguriert (zum Beispiel in `tooltip.axisPointer`).

Die folgenden Konfigurationen gibt es jedoch nur im globalen `axisPointer`: `axisPointer.triggerOn`, `axisPointer.link`.

#### Wie man `axisPointer` anzeigt:

In kartesischen (grid), polaren (polar) und Einzelachsen-Diagrammen (single axis) hat jede Achse ihren eigenen `axisPointer`.

Diese `axisPointer` werden standardmäßig nicht angezeigt, bis sie wie folgt konfiguriert wurden:

- Man setzt `eineAchse.axisPointer.show` (z.B. `xAxis.axisPointer.show`) auf `true`. Dann wird der `axisPointer` dieser Achse angezeigt.
- Man setzt `tooltip.trigger` auf `'axis'` oder `tooltip.axisPointer.type` auf `'cross'`. Dann wählt das Koordinatensystem automatisch die Achsen, die ihren `axisPointer` anzeigen. (`tooltip.axisPointer.axis` kann verwendet werden, um die Entscheidung zu ändern.) Es ist anzumerken, dass `axis.axisPointer` die Einstellungen von `tooltip.axisPointer` überschreibt.

#### Wie man die Beschriftung von `axisPointer` anzeigt:

Die Beschriftung des `axisPointer`s wird standardmäßig nicht angezeigt (nur die Referenzlinie wird standardmäßig angezeigt), bis sie wie folgt konfiguriert wurde:

- Man setzt `eineAchse.axisPointer.label.show` (z.B. `xAxis.axisPointer.label.show`) auf `true`. Dann wird die Beschriftung des `axisPointer`s angezeigt.
- Man setzt `tooltip.axisPointer.type` auf `'cross'`. Dann werden die Beschriftungen der gekreuzten `axisPointer` angezeigt.

#### Wie man `axisPointer` auf Geräten mit Berührungssteuerung konfiguriert:

Man setzt `eineAchse.axisPointer.handle.show` (z.B. `xAxis.axisPointer.handle.show`) auf `true`. Dann wird der ziehbare Knopf angezeigt. (Diese Feature wird für polar nicht unterstützt.)

**Hinweis:** Funktioniert der Tooltip in diesem Fall nicht gut, kann man versuchen, `tooltip.triggerOn` auf `'none'` zu setzen (für den Effekt: Zeige den Tooltip wenn der Finger den Knopf hält und verstecke ihn wenn der Finger den Knopf loslässt) oder `tooltip.alwaysShowContent` auf `true` zu setzen (dann wird der Tooltip immer angezeigt).

#### Auf Punkte einrasten:

Bei Zeit- und kontinuierlichen Achsen rastet der `axisPointer` automatisch auf Datenpunkten ein, falls `snap` auf `true` gesetzt wurde.

## 6.3.2 `chart.axisPointer.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.3.3 chart.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der `axisPointer` wird standardmäßig nicht angezeigt. Wurde `tooltip.trigger` jedoch auf 'axis' oder `tooltip.axisPointer.type` auf 'cross' gesetzt, so wird der `axisPointer` automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre `axisPointer` anzeigen. `tooltip.axisPointer.axis` kann verwendet werden, um die Wahl zu verändern.

### 6.3.4 chart.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.3.5 chart.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.3.6 chart.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.3.7 chart.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.3.8 chart.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.3.9 chart.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 6.3.10 chart.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

**Rückgabewert:**

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.3.11 chart.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 6.3.12 chart.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.3.13 chart.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 6.3.14 chart.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default -Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- norma
- bold
- bolder
- lighter
- {integer}

### 6.3.15 chart.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.3.16 chart.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.3.17 chart.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.3.18 chart.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.



### 6.3.19 chart.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.3.20 chart.axisPointer.label.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.3.21 chart.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.3.22 chart.axisPointer.label.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.3.23 chart.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.3.24 chart.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.3.25 chart.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.3.26 chart.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```

// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]

```

### 6.3.27 chart.axisPointer.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig gleich zu axis.axisLine.lineStyle.color .

### 6.3.28 chart.axisPointer.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.3.29 chart.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite der Beschriftung.

### 6.3.30 chart.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.3.31 chart.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.3.32 chart.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.3.33 chart.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.3.34 chart.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type auf 'line' steht.

### 6.3.35 chart.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.3.36 chart.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.3.37 chart.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Mögliche Wert:

- 'solid'
- 'dashed'
- 'dotted'

Erlaubte Werte

- solid
- dotted
- dashed

### 6.3.38 chart.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.3.39 chart.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.3.40 chart.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.3.41 chart.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.3.42 chart.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.3.43 chart.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.3.44 chart.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.3.45 chart.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.3.46 chart.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.3.47 chart.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.3.48 chart.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.3.49 chart.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.3.50 chart.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.3.51 chart.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.3.52 chart.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 6.3.53 chart.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.3.54 chart.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um den Handle zu verwenden.

### 6.3.55 chart.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon des Handles.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.3.56 chart.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.3.57 chart.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.3.58 chart.axisPointer.handle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.3.59 chart.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.3.60 chart.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.3.61 chart.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.3.62 chart.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.3.63 chart.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.3.64 chart.axisPointer.link

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

axisPointer können miteinander verlinkt werden. Der Begriff "link" zeigt an, dass Achsen miteinander synchronisiert sind und sich zusammen bewegen. Achsen Achsen werden anhand des Wertes von axisPointer verlinkt.

link ist ein Array, in dem jedes Element eine "link group" repräsentiert. Achsen werden verlinkt wenn sie in der selben link group referenziert werden. Zum Beispiel:

```
link: [
  {
    // Alle Achsen mit xAxisIndex 0, 3, 4 und yAxisName "sameName" werden
    verlinkt
    xAxisIndex: [0, 3, 4],
    yAxisName: 'sameName'
  },
  {
    // Alle Achsen mit xAxisId 'aa', 'cc' und alle angleAxis werden verlinkt.
    xAxisId: ['aa', 'cc'],
    angleAxis: 'all'
  },
  ...
]
```

Wie oben gezeigt können Achsen in einer link group auf diese Arten referenziert werden:

```
{
  // 'some' steht für den Dimensionsnamen einer Achse, namentlich 'x', 'y',
  // 'radius', 'angle' oder 'single'
  someAxisIndex: [...], // kann ein Array, Wert oder 'all' sein
  someAxisName: [...], // kann ein Array, Wert oder 'all' sein
  someAxisId: [...], // kann ein Array, Wert oder 'all' sein
}
```

### Wie verlinkt man Achsen mit verschiedenem axis.type?

Zum Beispiel, der Typ von axisA ist 'category' und der Typ von axisB ist 'time'. Wir können eine Konvertierungsfunktion (Mapper) in der link group schreiben um Werte von einer Achse zur anderen zu konvertieren. zum Beispiel:

```
link: [{
  xAxisIndex: [0, 1],
  yAxisName: ['yy'],
  mapper: function (sourceVal, sourceAxisInfo, targetAxisInfo) {
    if (sourceAxisInfo.axisName === 'yy') {
      // von Zeitstempel zu '2012-02-05'
      return echarts.format.formatTime('yyyy-MM-dd', sourceVal);
    }
    else if (targetAxisInfo.axisName === 'yy') {
      // von '2012-02-05' zu date
      return echarts.number.parseDate(dates[sourceVal]);
    }
    else {
      return sourceVal;
    }
  }
}]
```

Eingabeparameter von Mapper:

{number} sourceVal

{Object} sourceAxisInfo enthält {axisDim, axisId, axisName, axisIndex, ...}

{Object} targetAxisInfo enthält {axisDim, axisId, axisName, axisIndex, ...}

Rückgabewert von Mapper:

{number} Das Ergebnis der Konvertierung

### 6.3.65 chart.axisPointer.triggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mousemove|click' ]

Konditionen unter denen der Tooltip auslöst. Mögliche Werte:

- 'mousemove'  
Löse aus wenn die Maus sich bewegt.
- 'click'  
Löse aus wenn die Maus klickt.
- 'mousemove|click'  
Löse aus wenn die Maus klickt und sich bewegt.
- 'none'  
Löse nicht durch bewegen und klicken der Maus aus.

Erlaubte Werte

- mousemove
- click
- mousemove|click
- none

### 6.3.66 chart.axisPointer.label.textBorderStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderStyle: [5, 10], textBorderDashOffset: 5 }
```

## Erlaubte Werte

- dashed
- dotted
- solid

### 6.3.67 chart.axisPointer.label.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit textBorderStyle können wir den Linienstil flexibler gestalten.

### 6.3.68 chart.axisPointer.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll



### 6.3.69 chart.axisPointer.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.3.70 chart.axisPointer.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.3.71 chart.axisPointer.lineStyle.join

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.3.72 chart.axisPointer.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Axis Pointer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

## 6.4 ECharts (5) Brush

### 6.4.1 chart.brush.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

brush ist eine Komponente zum Auswählen von Bereichen, mit der Nutzer Teile der Daten eines Diagramms auswählen können, um diese im Detail zu betrachten oder Berechnungen mit ihnen durchzuführen.

### 6.4.2 chart.brush.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.4.3 chart.brush.toolbox

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: ['rect', 'polygon', 'keep', 'clear'] ]

In der Toolbox zu verwendende Knöpfe.

Knöpfe der Toolbox die zu brush gehören sind:

- 'rect': Für rechteckige Auswahlbox;
- 'polygon': Für vieleckige Auswahlbox;
- 'lineX': Für horizontale Auswahlbox;
- 'lineY': Für vertikale Auswahlbox;
- 'keep': Für das Wechseln zwischen einzelner und multipler Auswahl. Die Erstere unterstützt das Löschen des Auswahlbereichs beim Mausklick, die Letztere unterstützt mehrere Auswahlbereiche.
- 'clear': Für das Löschen aller Auswahlbereiche

### Erlaubte Werte

- rect
- polygon
- lineX
- lineY
- keep
- clear

## 6.4.4 chart.brush.brushLink

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Verbindet Interaktionen zwischen ausgewählten Elementen in verschiedenen Serien.

Es folgt ein Beispiel für das Aktivieren vom ausgewählten Effekt für Streu- und Paralleldiagramme sobald ein Streudiagramm ausgewählt ist.

brushLink ist ein Array aus seriesIndex-Werten, welches die Serien bestimmt, deren Interaktionen verbunden sind. Der Wert kann zum Beispiel sein:

- [3, 4, 5] für Interaktionen zwischen den Serien mit seriesIndex 3, 4 oder 5;
- 'all' für Interaktionen zwischen allen Serien;
- 'none', null oder undefined um brushLink zu deaktivieren.

### Achtung

brushLink ist ein Mapping von dataIndex. **Also sollten die Daten aller mit brushLink verbundenen Serien garantiert zueinander korrespondieren.**

Beispiel:

```
option = {
  brush: {
    brushLink: [0, 1]
  },
  series: [
    {
      type: 'bar'
      data: [232, 4434, 545, 654] // data hat 4 Einträge
    },
    {
      type: 'parallel',
      data: [[4, 5], [3, 5], [66, 33], [99, 66]] // data hat auch 4 Einträge,
      was zu dem data oben korrespondiert
    }
  ]
};
```

### 6.4.5 chart.brush.seriesIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'all' ]

Legt fest, welche der Serien brush-Auswahl verwenden kann. Der Wert kann sein:

- 'all': Alle Serien;
- Array: Array an Serienindices
- Nummer: Bestimmter Serienindex

### 6.4.6 chart.brush.geoIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der geo-Diagramme die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von brush.geoIndex, brush.xAxisIndex und brush.yAxisIndex festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:

```
option = {
  geo: {
    ...
  },
  brush: {
    geoIndex: 'all', // brush-Auswahl ist nur in den oberen geo-Diagrammen
    aktiviert
    ...
  }
};
```

## 6.4.7 chart.brush.xAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der xAxisIndex die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von brush.geoIndex, brush.xAxisIndex und brush.yAxisIndex festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:

```
option = {
  grid: [
    {...}, // grid 0
    {...} // grid 1
  ],
  xAxis: [
    {gridIndex: 1, ...}, // xAxis 0 für grid 1
    {gridIndex: 0, ...} // xAxis 1 für grid 0
  ],
  yAxis: [
    {gridIndex: 1, ...}, // yAxis 0 für grid 1
    {gridIndex: 0, ...} // yAxis 1 für grid 0
  ],
  brush: {
    xAxisIndex: [0, 1], // Brush-Auswahl ist nur für Koordinaten mit xAxisIndex 0
oder 1 aktiviert
    ...
  }
};
```

## 6.4.8 chart.brush.yAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Legt fest, welche der yAxisIndex die Brush-Auswahl benutzen können.

brush ist entweder *global* oder *gehört zu bestimmten Koordinaten*.

### Globale Brushes

Die Auswahl ist in diesem Fall überall in der ECharts-Instanz aktiviert. Dies ist die standardmäßige Situation, wenn brush nicht auf global gesetzt wurde.

### Koordinaten-Brushes

Die Auswahl ist in diesem Fall nur für die festgelegten Koordinaten aktiviert. Auswahlboxen werden anhand der Skalierung und Übersetzung von Koordinaten verändert (siehe roam und dataZoom).

In der Praxis sieht man oft Koordinaten-Brushes als häufiger getroffene Entscheidung, vor allem in geo-Diagrammen.

Die Achsen, für die die Auswahl aktiviert ist, lassen sich durch das Setzen von brush.geoIndex, brush.xAxisIndex und brush.yAxisIndex festlegen. Die Werte können sein:

- 'all': Für alle Achsen;
- Nummer, wie 0, für eine bestimmte Koordinate mit diesem Index;
- Array, wie [0, 4, 2], für Koordinaten mit diesen Indices;
- 'none', null oder undefined: Für keine Festlegung.

Beispiel:



```

option = {
  grid: [
    {...}, // grid 0
    {...} // grid 1
  ],
  xAxis: [
    {gridIndex: 1, ...}, // xAxis 0 für grid 1
    {gridIndex: 0, ...} // xAxis 1 für grid 0
  ],
  yAxis: [
    {gridIndex: 1, ...}, // yAxis 0 für grid 1
    {gridIndex: 0, ...} // yAxis 1 für grid 0
  ],
  brush: {
    yAxisIndex: [0, 1], // Brush-Auswahl ist nur für Koordinaten mit yAxisIndex 0
oder 1 aktiviert
    ...
  }
};

```

### 6.4.9 chart.brush.brushType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rect' ]

Standardtyp der Auswahl.

- 'rect': Für eine rechteckige Auswahlbox;
- 'polygon': Für eine vieleckige Auswahlbox;
- 'lineX': Für eine horizontale Auswahlbox;
- 'lineY': Für eine vertikale Auswahlbox;

#### Erlaubte Werte

- rect
- polygon
- lineX
- lineY

### 6.4.10 chart.brush.brushMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'single' ]

Standardmäßiger brush-Modus, dessen Wert sein kann:

- 'single': Für Einzelauswahl;
- 'multiple': Für Mehrfachauswahl.

Erlaubte Werte

- single
- multiple

### 6.4.11 chart.brush.transformable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Bestimmt, ob eine ausgewählte Box in ihrer Form verändert oder übersetzt werden kann.

### 6.4.12 chart.brush.brushStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Standardmäßiger brush-Stil, dessen Wert ist:

```
{
  borderWidth: 1,
  color: 'rgba(120,140,180,0.3)',
  borderColor: 'rgba(120,140,180,0.8)'
},
```

### 6.4.13 chart.brush.throttleType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'fixRate' ]

Standardmäßig wird brushSelected ausgelöst wann immer eine Auswahlbox ausgewählt oder bewegt wird, um die Umgebung über das Event zu informieren.

Zu häufiges Auslösen des Events kann jedoch zu Effizienzproblemen oder Beeinträchtigung der Animation führen. Deshalb stellt die brush-Komponente brush.throttleType und brush.throttleDelay zur Verfügung, um dieses Problem zu lösen.

Gültige Werte für throttleType sind:

- 'debounce': Um das Event erst auszulösen, wenn die Handlung aufgehört hat (keine weiteren Aktionen in einem bestimmten Zeitraum). Die Zeitschwelle kann über brush.throttleDelay gesetzt werden.
- 'fixRate': Um das Event mit einer bestimmten Frequenz auszulösen. Die Frequenz kann über brush.throttleDelay gesetzt werden.

### 6.4.14 chart.brush.throttleDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Brush

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

0 um Drosselung zu deaktivieren.

Standardmäßig wird brushSelected ausgelöst wann immer eine Auswahlbox ausgewählt oder bewegt wird, um die Umgebung über das Event zu informieren.

Zu häufiges Auslösen des Events kann jedoch zu Effizienzproblemen oder Beeinträchtigung der Animation führen. Deshalb stellt die brush-Komponente brush.throttleType und brush.throttleDelay zur Verfügung, um dieses Problem zu lösen.

Gültige Werte für throttleType sind:

- 'debounce': Um das Event erst auszulösen, wenn die Handlung aufgehört hat (keine weiteren Aktionen in einem bestimmten Zeitraum). Die Zeitschwelle kann über brush.throttleDelay gesetzt werden.
- 'fixRate': Um das Event mit einer bestimmten Frequenz auszulösen. Die Frequenz kann über brush.throttleDelay gesetzt werden.

#### 6.4.15 chart.brush.removeOnClick

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Löschen aller Auswahlboxen beim Klick aktiviert ist wenn brush.brushMode auf 'single' steht.

#### 6.4.16 chart.brush.inBrush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert visuelle Effekte für Elemente in der Auswahl.

Verfügbare visuelle Effekte sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Farbhelligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbwert in HSL.

In den meisten Fällen kann inBrush ungesetzt bleiben, wodurch die Standardkonfiguration verwendet wird.

#### 6.4.17 chart.brush.outOfBrush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert visuelle Effekte für Elemente außerhalb der Auswahl.

Verfügbare visuelle Effekte sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Farbhelligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbwert in HSL.

**Hinweis:** Ist outOfBrush ungesetzt, so wird die Farbe standardmäßig auf '#ddd' gesetzt. Ist die Farbe nicht erwünscht kann man folgendes verwenden:

```
brush: {
  ...,
  outOfBrush: {
    colorAlpha: 0.1
  }
}
```

### 6.4.18 chart.brush.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Brush
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10000 ]

z-Index der brush-Titelbox. Kann angepasst werden, wenn fehlerhafte Überlappungen auftreten.

## 6.5 ECharts (5) Calendar

### 6.5.1 chart.calendar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kalenderkoordinatensystem.

In ECharts können durch die Verwendung von Koordinatensystemen für die Erstellung des Kalenderdiagramms diese auch in Heatmaps, Streudiagrammen mit und ohne Effekte, oder Graphen verwendet werden.

#### **Kalenderlayout**

Kalenderkoordinatensysteme können horizontal oder vertikal platziert werden. Von der Konvention her ist der Heatmap-Kalender horizontal. Benötigen wir in anderen Fällen aber eine höhere Zellgröße, kann die Gesamtbreite zu hoch sein. In diesem Fall hilft `calendar.orient`.

### Anpassung an Containergröße

Kalenderkoordinatensysteme können dazu konfiguriert werden, sich an die Containergröße anzupassen, was nützlich ist, wenn die Seitengröße unbekannt ist. Zum einen kann, wie bei anderen Komponenten, die Platzierung und Größe am Kalender festgelegt werden: `left` / `right` / `top` / `bottom` / `width` / `height`, welche es dem Kalender ermöglichen, seine Größe anhand der Containergröße zu modifizieren. Außerdem kann `cellSize` spezifiziert werden, um die Größe der Kalenderzellen festzulegen.

## 6.5.2 `chart.calendar.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 6.5.3 `chart.calendar.zlevel`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

`zlevel`-Wert aller grafischen Elemente im Container.

`zlevel` wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen `zlevel`-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates `zlevel` legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plaziert.

## 6.5.4 chart.calendar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

## 6.5.5 chart.calendar.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Kalender-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {percent}
- {integer}
- left



- center
- right

## 6.5.6 chart.calendar.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Kalender-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

## 6.5.7 chart.calendar.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Kalender-Komponente und der rechten Seite des Containers

Der Wert von `right` kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}
- auto

### 6.5.8 `chart.calendar.bottom`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Kalender-Komponente und der unteren Seite des Containers.

Der Wert von `bottom` kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}

### 6.5.9 `chart.calendar.width`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Breite des Kalenderkoordinatensystems.

Hinweis: cellSize ist standardmäßig 20. Ist width gesetzt, wird cellSize[0] auf 'auto' gesetzt

### 6.5.10 chart.calendar.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Höhe des Kalenderkoordinatensystems.

Hinweis: cellSize ist standardmäßig 20. Ist height gesetzt, wird cellSize[1] auf 'auto' gesetzt.

### 6.5.11 chart.calendar.range

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Benötigt, Spannweite des Kalenderkoordinatensystems, unterstützt mehrere Formate.

Beispiele:

```
// Ein Jahr
range: 2017

// Ein Monat
range: '2017-02'

// Eine Spanne
range: ['2017-01-02', '2017-02-23']

// Hinweis: Diese Spanne wird interpretiert als ['2017-01-01', '2017-02-01']
range: ['2017-01', '2017-02']
```

## 6.5.12 chart.calendar.cellSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Die Größe der Kalenderzellen. Kann als einzelner Wert oder als Array gesetzt werden, in dem das erste Element die Breite und das zweite die Höhe ist.

Unterstützt Selbstanpassung: 'auto', die standardmäßige Höhe und Breite ist 20.

Beispiele:

```
// Setze Breite und Höhe auf 20
cellSize: 20

// Setze Breite auf 20, und Höhe auf 40
cellSize: [20, 40]

// Setze Breite und Höhe auf 40
cellSize: [40]

// Setze Breite und Höhe auf Selbstanpassung
cellSize: 'auto'

// Setze Breite auf Selbstanpassung, und Höhe auf 40
cellSize: ['auto', 40]
```

### 6.5.13 chart.calendar.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Ausrichtung des Kalenders.

Erlaubte Werte

- horizontal
- vertical

### 6.5.14 chart.calendar.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinie des Kalenderkoordinatensystems.

### 6.5.15 chart.calendar.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen.

### 6.5.16 chart.calendar.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 6.5.17 chart.calendar.splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Trennlinienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.5.18 chart.calendar.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.5.19 chart.calendar.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.5.20 chart.calendar.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:



```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.5.21 chart.calendar.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.5.22 chart.calendar.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.5.23 chart.calendar.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.5.24 chart.calendar.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.5.25 chart.calendar.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil jedes Rechtecks im Kalenderkoordinatensystem.

### 6.5.26 chart.calendar.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Kalenderfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.5.27 chart.calendar.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: '#ccc' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 6.5.28 chart.calendar.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 6.5.29 chart.calendar.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 6.5.30 chart.calendar.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.5.31 chart.calendar.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.5.32 chart.calendar.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.5.33 chart.calendar.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.5.34 chart.calendar.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.5.35 chart.calendar.dayLabel.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Tage im Kalenderkoordinatensystem.

### 6.5.36 chart.calendar.dayLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tagesbeschriftungen angezeigt werden sollen.

### 6.5.37 chart.calendar.dayLabel.firstDay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Welcher Tag im Kalender als erstes aufgeführt wird, standardmäßig Sonntag.

Beispiel:



```
calendar: [{
  dayLabel: {
    firstDay: 1 // Starte mit Montag
  }
}]
```

### 6.5.38 chart.calendar.dayLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Abstand zwischen Tagesbeschriftung und Achsenlinie.

### 6.5.39 chart.calendar.dayLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position der Tagesbeschriftungen, am Anfang oder am Ende des Diagramms.

#### Erlaubte Werte

- start
- end

### 6.5.40 chart.calendar.dayLabel.nameMap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'en' ]

Bezeichnungen der Wochentage, standardmäßig 'en'. Unterstützt Chinesisch, Englisch und benutzerdefiniert. Index 0 ist immer Sonntag.

Beispiele:

```
// Abkürzung für Englisch ['S', 'M', 'T', 'W', 'T', 'F', 'S']
nameMap: 'en'

// Abkürzung für Chinesisch ['?', '?', '?', '?', '?', '?', '?']
nameMap: 'cn'

// Benutzerdefiniert: Englisch und Chinesisch gemischt, oder leer
nameMap: ['S', '?', 'T', '?', '?', '?', 'S']

calendar: [{
  dayLabel: {
    nameMap: 'en'
  }
}]
```

### 6.5.41 chart.calendar.dayLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Textfarbe.

## 6.5.42 chart.calendar.dayLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.5.43 chart.calendar.dayLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.5.44 chart.calendar.dayLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.5.45 chart.calendar.dayLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.5.46 chart.calendar.dayLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.5.47 chart.calendar.dayLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

## 6.5.48 chart.calendar.dayLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.5.49 chart.calendar.dayLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	calendar

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.5.50 chart.calendar.dayLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.5.51 chart.calendar.dayLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.5.52 chart.calendar.dayLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.5.53 chart.calendar.dayLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].



Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.5.54 chart.calendar.dayLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.5.55 chart.calendar.dayLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.5.56 chart.calendar.dayLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.5.57 chart.calendar.dayLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.5.58 chart.calendar.dayLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.5.59 chart.calendar.dayLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.5.60 chart.calendar.dayLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.5.61 chart.calendar.dayLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.5.62 chart.calendar.dayLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.5.63 chart.calendar.dayLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.5.64 chart.calendar.dayLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.5.65 chart.calendar.dayLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.5.66 chart.calendar.dayLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.5.67 chart.calendar.monthLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Monatsbeschriftung im Kalenderkoordinatensystem.

## 6.5.68 chart.calendar.monthLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Monatsbeschriftungen angezeigt werden sollen.

## 6.5.69 chart.calendar.monthLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 6.5.70 chart.calendar.monthLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Der Abstand zwischen den Monatsbeschriftungen und der Achsenlinie.

### 6.5.71 chart.calendar.monthLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position der Monatsbeschriftungen, am Anfang oder am Ende des Diagramms.

Erlaubte Werte

- start
- end

### 6.5.72 chart.calendar.monthLabel.nameMap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'en' ]

Bezeichnungen der Monate, standardmäßig 'en'. Unterstützt Chinesisch, Englisch und benutzerdefiniert. Index 0 ist immer Januar.

Beispiele:

```
// Abkürzung für Englisch [
    'Jan', 'Feb', 'Mar',
    'Apr', 'May', 'Jun',
    'Jul', 'Aug', 'Sep',
    'Oct', 'Nov', 'Dec'
],
nameMap: 'en'

// Abkürzung für Chinesisch [
    '??', '??', '??',
    '??', '??', '??',
    '??', '??', '??',
    '??', '???'', '???'
]
nameMap: 'cn'

// Benutzerdefiniert: Englisch und Chinesisch gemischt oder leer
nameMap: [
    '??', 'Feb', '??',
    '??', 'May', '??',
    '??', '??', '',
    '??', 'Nov', '???'
],

calendar: [{
  monthLabel: {
    nameMap: 'en'
  }
}]
```

### 6.5.73 chart.calendar.monthLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Monatsbeschriftungen, unterstützt String-Templates und Callback-Funktionen.

Beispiele:

```
// Verwendung von String-Templates; z.B. 2017-02
/*
  Template-Variablen:
  {nameMap} standardmäßiges nameMap, z.B. 'Feb'
  {yyyy}   Vierstelliges Jahr, z.B. 2017
  {yy}    Zweistelliges Jahr, z.B. 17
  {MM}    Zweistelliger Monat, z.B. 02
  {M}     Einstelliger Monat, z.B. 2
*/
formatter: '{yyyy}-{MM}'

// Verwendung von Callback-Funktionen;
/*
  Funktionsparameter:
  param.nameMap standardmäßiges nameMap, z.B. 'Feb'
  param.yyyy   Vierstelliges Jahr, z.B. 2017
  param.yy    Zweistelliges Jahr, z.B. 17
  param.MM    Zweistelliger Monat, z.B. 02
  param.M     Einstelliger Monat, z.B. 2
*/
formatter: function (param) {
  // ...
  return param.MM;
}
```

### 6.5.74 chart.calendar.monthLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Textfarbe.

### 6.5.75 chart.calendar.monthLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.5.76 chart.calendar.monthLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.5.77 chart.calendar.monthLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.5.78 chart.calendar.monthLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.5.79 chart.calendar.monthLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

Erlaubte Werte

- top
- middle
- bottom

### 6.5.80 chart.calendar.monthLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.5.81 chart.calendar.monthLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.5.82 chart.calendar.monthLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.5.83 chart.calendar.monthLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.5.84 chart.calendar.monthLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.5.85 chart.calendar.monthLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.5.86 chart.calendar.monthLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.5.87 chart.calendar.monthLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.5.88 chart.calendar.monthLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.5.89 chart.calendar.monthLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.5.90 chart.calendar.monthLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.5.91 chart.calendar.monthLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.5.92 chart.calendar.monthLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.5.93 chart.calendar.monthLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.5.94 chart.calendar.monthLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.5.95 chart.calendar.monthLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.5.96 chart.calendar.monthLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.5.97 chart.calendar.monthLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.5.98 chart.calendar.monthLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.5.99 chart.calendar.yearLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Jahresbeschriftung im Kalenderkoordinatensystem.

### 6.5.100 chart.calendar.yearLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Jahresbeschriftungen angezeigt werden sollen.

### 6.5.101 chart.calendar.yearLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Der Abstand zwischen der Jahresbeschriftung und der Achsenlinie.

### 6.5.102 chart.calendar.yearLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Jahresbeschriftungen.

Standardwert: Wenn orient auf 'horizontal' steht, ist es 'left', wenn orient auf 'vertical' steht, ist es 'top'.

Erlaubte Werte

- top
- bottom
- left
- right

### 6.5.103 chart.calendar.yearLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Jahresbeschriftungen, unterstützt String-Templates und Callback-Funktionen.

Zeigt standardmäßig den Spann des momentanen Jahres an, oder, falls der Intervall über Jahre hinweg geht, das erste und das letzte Jahr.

Beispiele:

```
// Verwendung von String-Templates, z.B. ['2017-10-11', '2018-01-21']
/*
  Template-Variablen:
  {nameMap} Standardmäßige nameMap, z.B. '2017-2018'
  {start}   Startjahr, z.B. 2017
  {end}    Endjahr, z.B. 2018
*/
formatter: '{start}-{end}'

// Verwendung von Callback-Funktionen
/*
  Funktionsparameter:
  param.nameMap   Standardmäßige nameMap, z.B. '2017-2018'
  param.start    Startjahr, z.B. 2017
  param.end      Endjahr, z.B. 2018
*/
formatter: function (param) {
  // ...
  return param.end;
}
```

#### 6.5.104 chart.calendar.yearLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

#### 6.5.105 chart.calendar.yearLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.5.106 chart.calendar.yearLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.5.107 chart.calendar.yearLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.5.108 chart.calendar.yearLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.5.109 chart.calendar.yearLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 6.5.110 chart.calendar.yearLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.5.111 chart.calendar.yearLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.5.112 chart.calendar.yearLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.5.113 chart.calendar.yearLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.5.114 chart.calendar.yearLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.5.115 chart.calendar.yearLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.5.116 chart.calendar.yearLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.5.117 chart.calendar.yearLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.5.118 chart.calendar.yearLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.5.119 chart.calendar.yearLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.5.120 chart.calendar.yearLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.5.121 chart.calendar.yearLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.5.122 chart.calendar.yearLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.5.123 chart.calendar.yearLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.5.124 chart.calendar.yearLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.5.125 chart.calendar.yearLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Calendar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.5.126 chart.calendar.yearLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.5.127 chart.calendar.yearLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.5.128 chart.calendar.yearLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.5.129 chart.calendar.yearLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.5.130 chart.calendar.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Calendar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

## 6.6 ECharts (5) Data

### 6.6.1 chart.data.hierachy.render

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: "default" ]

Wird für Sunburst-Diagramme verwendet, bei denen die Hierarchie der Daten festgelegt werden muss.

Zum Beispiel:

```
dashlet.addOption("chart.data.hierachy.render", "all");
```

#### Erlaubte Werte

- all

### 6.6.2 chart.dynamicSum.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Anzeige einer dynamischen Summe in einem Diagramm (Chart)

```
dashlet.addOption("chart.dynamicSum.show", "true");
```

### 6.6.3 chart.dynamicSum.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die Position des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.position", "top");
```

### 6.6.4 chart.dynamicSum.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die Farbe des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.color", "#000");
```

### 6.6.5 chart.dynamicSum.roundedDigits

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Aktiviert die Möglichkeit, die roundedDigits des dynamicSum-Werts in einem Diagramm festzulegen

```
dashlet.addOption("chart.dynamicSum.roundedDigits", "2");
```

### 6.6.6 chart.dynamicSum.excludeSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einige Serien von der dynamicSum-Berechnung in einem Diagramm auszuschließen

```
dashlet.addOption("chart.dynamicSum.excludeSeriesList","Motor City Art Classics;Unimax Art Galleris;Highway 66 Mini Classics");
```

### 6.6.7 chart.dynamicSum.retainStyleSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, den Stil einiger Serien beizubehalten, wenn die dynamicSum-Berechnung in einem Diagramm verwendet wird

```
dashlet.addOption("chart.dynamicSum.retainStyleSeriesList","Motor City Art Classics;Unimax Art Galleris");
```

### 6.6.8 chart.dynamicSum.retainStyleSeriesList

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, den Stil einiger Serien beizubehalten, wenn die dynamicSum-Berechnung in einem Diagramm verwendet wird

```
dashlet.addOption("chart.dynamicSum.retainStyleSeriesList","Motor City Art Classics;Unimax Art Galleris");
```

### 6.6.9 chart.dynamicSum.customSeriesHandlerBlock

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einen benutzerdefinierten Code für die seriesHandler-Funktion einzufügen, um die dynamicSum in einem Diagramm zu berechnen

```
dashlet.addOption("chart.dynamicSum.customSeriesHandlerBlock","return null");
```



## 6.6.10 chart.dynamicSum.customHandleLegendSelectChangedBlock

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.7
<b>Schlagworte / Tags</b>	

Ermöglicht die Möglichkeit, einen benutzerdefinierten Code für die LegendSelectChanged -Funktion einzufügen, um die dynamicSum in einem Diagramm zu berechnen

```
dashlet.addOption("chart.dynamicSum.customHandleLegendSelectChangedBlock", "return null");
```

## 6.7 ECharts (5) Data Zoom

### 6.7.1 chart.dataZoom.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die dataZoom-Komponente wird verwendet um auf einem spezifischen Bereich zu zoomen, wodurch Nutzer Daten genauer untersuchen, sich einen Überblick über die Daten verschaffen oder Ausreißer eliminieren können.

Diese Typen der dataZoom-Komponente werden unterstützt:

- dataZoomInside: Die Zoom-Funktionalität ist in das Koordinatensystem eingebettet, wodurch Nutzer durch Mausbewegungen oder Fingerberührungen (bei Touchscreens) das Koordinatensystem durchwandern und zoomen können.
- dataZoomSlider: Ein spezieller Schieberegler wird bereitgestellt, durch den Koordinatensysteme durch Mausbewegungen oder Fingerberührungen (bei Touchscreens) durchwandert oder gezoomt werden können.
- dataZoomSelect: Ein Marquee-Tool wird bereitgestellt, um das Koordinatensystem zu durchwandern oder zu zoomen. Dieses ist toolbox.feature.dataZoom, welches nur in toolbox konfiguriert werden kann.

## 6.7.2 chart.dataZoom.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Typ der dataZoom-Komponente.

Erlaubte Werte

- inside
- slider

## 6.7.3 chart.dataZoom.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 6.7.4 chart.dataZoom.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: true ]

Spezifiziert, ob die Komponente angezeigt werden soll. Steht der Wert auf false, wird sie nicht angezeigt, aber ihre Funktionen zum Filtern von Daten funktionieren trotzdem.

### 6.7.5 chart.dataZoom.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(47, 69, 84, 0)' ]

Die Hintergrundfarbe der Komponente.

### 6.7.6 chart.dataZoom.dataBackground.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil des Schattens der Daten

### 6.7.7 chart.dataZoom.dataBackground.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil des Schattens.

### 6.7.8 chart.dataZoom.dataBackground.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #2f4554 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.9 chart.dataZoom.dataBackground.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0.5 ]

Linienbreite.

## 6.7.10 chart.dataZoom.dataBackground.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

## 6.7.11 chart.dataZoom.dataBackground.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.7.12 chart.dataZoom.dataBackground.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.7.13 chart.dataZoom.dataBackground.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.14 chart.dataZoom.dataBackground.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.15 chart.dataZoom.dataBackground.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.16 chart.dataZoom.dataBackground.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Schattenbereichs.

### 6.7.17 chart.dataZoom.dataBackground.areaStyle.color

<b>Datentyp</b>	COLOR
-----------------	-------



<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(47,69,84,0.3) ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.18 chart.dataZoom.dataBackground.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.7.19 chart.dataZoom.dataBackground.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 6.7.20 chart.dataZoom.dataBackground.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.21 chart.dataZoom.dataBackground.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.22 chart.dataZoom.dataBackground.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.23 chart.dataZoom.fillerColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(47,69,84,0.25)' ]

Die Farbe, die den ausgewählten Bereich füllt.

### 6.7.24 chart.dataZoom.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ddd' ]

Die Farbe des Rahmens.

### 6.7.25 chart.dataZoom.handleIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'M8.2,13.6V3.9H6.3v9.7H3.1v14.9h3.3v9.7h1.8v-9.7h3.3V13.6H8.2z M9.7,24.4H4.8v-1.4h4.9V24.4z M9.7,19.1H4.8v-1.4h4.9V19.1z' ]

Bildliche Form des Handles, welche Pfadstrings unterstützt.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.7.26 chart.dataZoom.handleSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die Größe des Kontrollhandles. Kann in Pixeln angegeben werden, oder in Prozentwerten relativ zur Breite der dataZoom-Komponente. Ist standardmäßig gleich zur Breite der dataZoom-Komponente.

Erlaubte Werte

- {integer}
- {percent}

### 6.7.27 chart.dataZoom.handleStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Handles.

### 6.7.28 chart.dataZoom.handleStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #a7b7cc ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.29 chart.dataZoom.handleStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.7.30 chart.dataZoom.handleStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.7.31 chart.dataZoom.handleStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- solid
- dashed
- dotted

## 6.7.32 chart.dataZoom.handleStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.7.33 chart.dataZoom.handleStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.7.34 chart.dataZoom.handleStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.35 chart.dataZoom.handleStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.36 chart.dataZoom.handleStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.37 chart.dataZoom.labelPrecision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Präzision der Beschriftung wenn in Zahlenform. Wird standardmäßig anhand der Datenanzahl bestimmt.

### 6.7.38 chart.dataZoom.labelFormatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Das Formatierwerkzeug für die Beschriftung.

- Ist eine Template, falls ein String angegeben wird. Zum Beispiel 'aaaa{value}bbbb', wobei {value} durch den tatsächlichen Datenwert ersetzt wird.
- Ist eine Callback-Funktion, falls eine Funktion angegeben wird. Zum Beispiel:

```

/**
 * @param {*} value Wenn axis.type 'category' ist, ist `value` der Index von
axis.data.
 *           Sonst ist `value` der momentane Wert.
 * @param {string} valueStr Innerer formatierter String.
 * @return {string} Gibt die formatierte Beschriftung zurück.
 */
labelFormatter: function (value, valueStr) {
  return 'aaa' + value + 'bbb';
}

```

### 6.7.39 chart.dataZoom.showDetail

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Details angezeigt werden sollen. Das heißt, ob detaillierte Informationen über die Daten während dem Ziehen angezeigt werden sollen.

### 6.7.40 chart.dataZoom.showDataShadow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Spezifiziert, ob der Schatten der Daten in der dataZoom-Komponente angezeigt werden soll, wodurch die Tendenz der Daten schnell angezeigt werden kann.

### 6.7.41 chart.dataZoom.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht beim Ziehen aktualisiert werden soll. Steht der Wert auf false, wird die Ansicht erst aktualisiert, wenn nicht mehr gezogen wird.

### 6.7.42 chart.dataZoom.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil der dataZoom-Komponente.

### 6.7.43 chart.dataZoom.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 6.7.44 chart.dataZoom.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.7.45 chart.dataZoom.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.7.46 chart.dataZoom.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.7.47 chart.dataZoom.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.7.48 chart.dataZoom.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.7.49 chart.dataZoom.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.



### 6.7.50 chart.dataZoom.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.7.51 chart.dataZoom.textStyle.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.7.52 chart.dataZoom.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.7.53 chart.dataZoom.textStyle.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.7.54 chart.dataZoom.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.7.55 chart.dataZoom.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.7.56 chart.dataZoom.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.7.57 chart.dataZoom.xAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche xAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Standardmäßig wird die erste xAxis kontrolliert, die parallel zum dataZoom liegt wenn dataZoom.orient auf 'horizontal' steht. Es wird aber empfohlen, den Wert explizit zu definieren.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```
option: {
  xAxis: [
    {...}, // Die erste xAxis
    {...}, // Die zweite xAxis
    {...}, // Die dritte xAxis
    {...} // Die vierte xAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      xAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                        // die erste und dritte xAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      xAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                  // die vierte xAxis kontrolliert
    }
  ]
}
```

## 6.7.58 chart.dataZoom.yAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche yAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Standardmäßig wird die erste yAxis kontrolliert, die parallel zum dataZoom liegt wenn dataZoom.orient auf 'vertical' steht. Es wird aber empfohlen, den Wert explizit zu definieren.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```

option: {
  yAxis: [
    {...}, // Die erste xAxis
    {...}, // Die zweite xAxis
    {...}, // Die dritte xAxis
    {...} // Die vierte xAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      yAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                        // die erste und dritte yAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      yAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                  // die vierte yAxis kontrolliert
    }
  ]
}

```

### 6.7.59 chart.dataZoom.radiusAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche radiusAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```

option: {
  radiusAxis: [
    {...}, // Die erste radiusAxis
    {...}, // Die zweite radiusAxis
    {...}, // Die dritte radiusAxis
    {...} // Die vierte radiusAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      radiusAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
                             // die erste und dritte radiusAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      radiusAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
                        // die vierte radiusAxis kontrolliert
    }
  ]
}

```

### 6.7.60 chart.dataZoom.angleAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, welche angleAxis vom dataZoom kontrolliert wird/werden, wenn ein polares Koordinatensystem benutzt wird.

Ist der Wert eine einzelne Zahl, so wird eine Achse kontrolliert. Ist der Wert ein Array, können mehrere Achsen kontrolliert werden.

Zum Beispiel:

```

option: {
  angleAxis: [
    {...}, // Die erste angleAxis
    {...}, // Die zweite angleAxis
    {...}, // Die dritte angleAxis
    {...} // Die vierte angleAxis
  ],
  dataZoom: [
    { // Die erste dataZoom-Komponente
      angleAxisIndex: [0, 2] // Zeigt an, dass diese dataZoom-Komponente
        // die erste und dritte angleAxis kontrolliert
    },
    { // Die zweite dataZoom-Komponente
      angleAxisIndex: 3 // Zeigt an, dass diese dataZoom-Komponente
        // die vierte angleAxis kontrolliert
    }
  ]
}

```

### 6.7.61 chart.dataZoom.filterMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'filter' ]

Im Allgemeinen zoomt und bewegt sich die dataZoom-Komponente durch das Koordinatensystem, indem die Daten gefiltert und die Fenster der Achsen intern gesetzt werden.

Das Verhalten variiert je nach ausgewählten Filtereinstellungen (dataZoom.filterMode).

Mögliche Werte:

- 'filter': Daten außerhalb des Fensters werden **gefiltert**, was zu Veränderungen der Fenster auf anderen Achsen führen kann. Jedes Datenelement wird dann gefiltert, wenn eine seiner relevanten Dimensionen außerhalb des Fensters liegt.
- 'weakFilter': Daten außerhalb des Fensters werden **gefiltert**, was zu Veränderungen der Fenster auf anderen Achsen führen kann. Jedes Datenelement wird dann gefiltert, wenn alle seiner relevanten Dimensionen außerhalb des Fensters liegen.
- 'empty': Daten außerhalb des Fensters werden **auf NaN gesetzt**, wodurch sich die Fenster der anderen Achsen nicht verändern.
- 'none': Daten nicht filtern.

Wie filterMode gesetzt wird ist Sache des Nutzers, und hängt von den Anforderungen und Szenarien ab. Empirisch:

- Wenn nur die X- oder die Y-Achse von dataZoom kontrolliert wird, wird typischerweise filterMode: 'filter' verwendet, wodurch die anderen Achsen ihre Fenster automatisch an die gefilterten Daten anpassen können.
- Wenn sowohl die X- als auch die Y-Achse von dataZoom kontrolliert wird :
  - Wenn die Achsen sich nicht gegenseitig beeinflussen sollen (z.B. in einem Streudiagramm mit beiden Achsen auf Typ 'value'), sollten sie auf filterMode: 'empty' gesetzt werden.
  - Wenn die X-Achse die Hauptachse und die Y-Achse die Hilfsachse ist (oder umgekehrt) (z.B. in einem Balkendiagramm, wenn dataZoomX gezogen wird um das Fenster der X-Achse zu verändern, soll die Y-Achse sich an den neuen Datenbereich anpassen, wenn aber dataZoomY gezogen wird, soll die X-Achse sich nicht verändern), in diesem Fall sollte die X-Achse auf filterMode: 'filter' gesetzt werden, während die Y-Achse auf filterMode: 'empty' gesetzt werden sollte.

Dies wird im folgenden Beispiel demonstriert:

```
option = {
  dataZoom: [
    {
      id: 'dataZoomX',
      type: 'slider',
      xAxisIndex: [0],
      filterMode: 'filter'
    },
    {
      id: 'dataZoomY',
      type: 'slider',
      yAxisIndex: [0],
      filterMode: 'empty'
    }
  ],
  xAxis: {type: 'value'},
  yAxis: {type: 'value'},
  series{
    type: 'bar',
    data: [
      // Die erste Spalte gehört zur X-Achse,
      // die zweite Spalte gehört zur Y-Achse.
      [12, 24, 36],
      [90, 80, 70],
      [3, 9, 27],
      [1, 11, 111]
    ]
  }
}
```

Im oberen Beispiel steht dataZoomX auf 'filter'. Wenn der Nutzer dataZoomX zieht (und dataZoomY nicht verändert), und der Wertebereich dadurch auf [2, 50] verändert wird, geht dataZoomX durch die erste Spalte von series.data, und filtert alle Zeilen raus, deren Wert in der ersten Spalte nicht im Bereich [2, 50] liegt. Das Ergebnis für series.data ist:



```
[
  [12, 24, 36],
  // [90, 80, 70] Dieses Element wird gefiltert, da 90 nicht im Fenster ist.
  [3, 9, 27]
  // [1, 11, 111] Dieses Element wird gefiltert, da 1 nicht im Fenster ist.
]
```

Vor dem Filtern hatte die zweite Spalte, welche zur Y-Achse gehört, die Werte 24, 80, 9, 11. Nach dem Filtern sind nur 24 und 9 übrig. Das Ausmaß der Y-Achse ändert sich, um sich an die zwei Werte anzupassen (falls `yAxis.min` und `yAxis.max` nicht gesetzt sind).

Es kann also `filterMode: 'filter'` verwendet werden, damit sich die anderen Achsen an gefilterte Daten anpassen können.

Betrachten wir das Beispiel nun erneut. `dataZoomY` ist auf `filterMode: 'empty'` gesetzt. Wenn der Nutzer `dataZoomY` zieht (und `dataZoomX` nicht verändert), und der Wertebereich dadurch auf `[10, 60]` verändert wird, geht `dataZoomY` durch die zweite Spalte von `series.data`, und setzt alle Werte auf `NaN`, die nicht im Bereich `[10, 60]` liegen (`NaN` führt dazu, dass grafische Elemente, z.B. Balkenelemente, nicht angezeigt werden, aber weiter ihren Platz halten). Das Ergebnis für `series.data` ist:

```
[
  [12, 24, 36],
  [90, NaN, 70], // Auf NaN gesetzt
  [3, NaN, 27], // Auf NaN gesetzt
  [1, 11, 111]
]
```

In diesem Fall ändert sich die erste Spalte (also 12, 90, 3, 1, welche zur X-Achse gehört) nicht. Das Ziehen der Y-Achse ändert also nicht das Fenster der X-Achse, was nützlich für z.B. das Eliminieren von Ausreißern ist.

### Erlaubte Werte

- filter
- weakFilter
- empty
- none

## 6.7.62 chart.dataZoom.start

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der prozentuale Startwert des Fensters des Datenbereichs, in der Spanne von 0 bis 100.  
dataZoom.start und dataZoom.end definieren das Datenfenster in **prozentualer** Form.

### 6.7.63 chart.dataZoom.end

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Der prozentuale Endwert des Fensters des Datenbereichs, in der Spanne von 0 bis 100.  
dataZoom.start und dataZoom.end definieren das Datenfenster in **prozentualer** Form.

### 6.7.64 chart.dataZoom.startValue

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der absolute Startwert des Fensters, funktioniert nicht, wenn dataZoom.end gesetzt ist.  
dataZoom.startValue und dataZoom.endValue definieren das Datenfenster in **absoluter** Form.

Es ist anzumerken, dass, falls eine Achse als Kategorieachse gesetzt ist, startValue als Index des Arrays axis.data oder als Wert des Arrays gesetzt werden kann. Im letzteren Fall wird der Wert automatisch zu einem Index übersetzt.

### 6.7.65 chart.dataZoom.endValue

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der absolute Endwert des Fensters, funktioniert nicht, wenn dataZoom.end gesetzt ist.

dataZoom.startValue und dataZoom.endValue definieren das Datenfenster in **absoluter** Form.

Es ist anzumerken, dass, falls eine Achse als Kategorieachse gesetzt ist, endValue als Index des Arrays axis.data oder als Wert des Arrays gesetzt werden kann. Im letzteren Fall wird der Wert automatisch zu einem Index übersetzt.

### 6.7.66 chart.dataZoom.minSpan

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die minimale Fenstergröße, in Prozent, mit einem Wert im Bereich von [0, 100].

Wenn dataZoom.minValueSpan gesetzt ist, funktioniert minSpan nicht mehr.

### 6.7.67 chart.dataZoom.maxSpan

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Begrenzt die maximale Fenstergröße, in Prozent, mit einem Wert im Bereich von [0, 100].

Wenn dataZoom.maxValueSpan gesetzt ist, funktioniert maxSpan nicht mehr.

### 6.7.68 chart.dataZoom.minValueSpan

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die minimale Fenstergröße.

Zum Beispiel: Eine Zeitachse kann auf  $3600 * 24 * 1000 * 5$  gesetzt werden, um "5 Tage" darzustellen. Eine Kategorieachse kann auf 5 gesetzt werden, um 5 Kategorien darzustellen.

### 6.7.69 chart.dataZoom.maxValueSpan

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Begrenzt die maximale Fenstergröße.

Zum Beispiel: Eine Zeitachse kann auf  $3600 * 24 * 1000 * 5$  gesetzt werden, um "5 Tage" darzustellen. Eine Kategorieachse kann auf 5 gesetzt werden, um 5 Kategorien darzustellen.

### 6.7.70 chart.dataZoom.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob das Layout der dataZoom-Komponente horizontal oder vertikal ist. Bestimmt außerdem, ob im kartesischen Koordinatensystem standardmäßig die horizontale oder vertikale Achse kontrolliert werden soll.

Erlaubte Werte

- horizontal
- vertical

### 6.7.71 chart.dataZoom.zoomLock

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Größe des Fensters (des ausgewählten Bereichs) festgelegt sein soll.

Steht der Wert auf true, ist die Größe des Fensters festgelegt. Das heißt, nur die Translation (durch Ziehen mit Maus oder Berührung) steht zur Verfügung, das Zoomen nicht.

### 6.7.72 chart.dataZoom.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Spezifiziert die Rate der Aktualisierung der Ansicht, in millisekunden (ms).

Steht animation auf true und animationDurationUpdate auf einem Wert größer 0, kann throttle auf dem Standardwert 100 gelassen werden (oder auf einen Wert größer 0 gesetzt werden). Die Animation beim Ziehen ist sonst möglicherweise nicht flüssig.

Steht animation auf false oder animationDurationUpdate auf 0, und ist die Datenanzahl nicht sehr groß, so kann die Animation beim Ziehen unflüssig wirken. In diesem Fall hilft es, throttle auf 0 zu setzen.

### 6.7.73 chart.dataZoom.rangeMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zum Beispiel: rangeMode: ['value', 'percent'] bedeutet, dass absolute Werte am Anfang und prozentuale Werte am Ende verwendet werden sollen.

Erlaubte Werte

- value
- percent

### 6.7.74 chart.dataZoom.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

### 6.7.75 chart.dataZoom.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.7.76 chart.dataZoom.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 6.7.77 chart.dataZoom.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

### 6.7.78 chart.dataZoom.right

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}
- auto

### 6.7.79 chart.dataZoom.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der dataZoom-slider-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}

## 6.7.80 chart.dataZoom.disabled

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der innere Zoom deaktiviert werden soll.

## 6.7.81 chart.dataZoom.zoomOnMouseWheel

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Wie das Zoomen ausgelöst wird. Mögliche Werte:

- true: Drehen des Mausekkrads löst Zoomen aus.
- false: Drehen des Mausekkrads löst kein Zoomen aus.
- 'shift': shift halten und Drehen des Mausekkrads löst Zoomen aus.
- 'ctrl': strg halten und Drehen des Mausekkrads löst Zoomen aus.
- 'alt': alt halten und Drehen des Mausekkrads löst Zoomen aus.

### Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

## 6.7.82 chart.dataZoom.moveOnMouseMove

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Wie die Bewegung des Datenfensters ausgelöst wird. Mögliche Werte:

- true: Bewegung der Maus löst Bewegung des Datenfensters aus.
- false: Bewegung der Maus löst keine Bewegung des Datenfensters aus.
- 'shift': shift halten und Maus bewegen löst Bewegung des Datenfensters aus.
- 'ctrl': strg halten und Maus bewegen löst Bewegung des Datenfensters aus.
- 'alt': alt halten und Maus bewegen löst Bewegung des Datenfensters aus.

Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

## 6.7.83 chart.dataZoom.moveOnMouseWheel

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Wie die Bewegung des Datenfensters ausgelöst wird. Mögliche Werte:

- true: Drehen des Mousrads löst Bewegung des Datenfensters aus.
- false: Drehen des Mousrads löst keine Bewegung des Datenfensters aus.
- 'shift': shift halten und Drehen des Mousrads löst Bewegung des Datenfensters aus.
- 'ctrl': strg halten und Drehen des Mousrads löst Bewegung des Datenfensters aus.

- 'alt': alt halten und Drehen des Mausems löst Bewegung des Datenfensters aus.

## Erlaubte Werte

- {boolean}
- shift
- ctrl
- alt

### 6.7.84 chart.dataZoom.preventDefaultMouseMove

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das standardmäßige Verhalten des 'Maus bewegt'-Events verhindert werden soll.

### 6.7.85 chart.dataZoom.dataBackground.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.7.86 chart.dataZoom.dataBackground.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.7.87 chart.dataZoom.dataBackground.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.7.88 chart.dataZoom.dataBackground.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.7.89 chart.dataZoom.handleStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.7.90 chart.dataZoom.handleStyle.borderCap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.7.91 chart.dataZoom.handleStyle.borderJoin

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.7.92 chart.dataZoom.handleStyle.borderMiterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.7.93 chart.dataZoom.textStyle.textBorderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit textBorderDashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel:



```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

### Erlaubte Werte

- dashed
- dotted
- solid

## 6.7.94 chart.dataZoom.textStyle.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit textBorderType können wir den Linienstil flexibler gestalten.

## 6.7.95 chart.dataZoom.textStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

### Erlaubte Werte

- truncate
- break
- breakAll

### 6.7.96 chart.dataZoom.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.7.97 chart.dataZoom.selectedDataBackground.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der Stil des selektierete Schattens der Daten

### 6.7.98 chart.dataZoom.selectedDataBackground.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Liniensstil des selektierete Schattens.

## 6.7.99 chart.dataZoom.selectedDataBackground.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: #2f4554 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.100 chart.dataZoom.selectedDataBackground.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0.5 ]

Linienbreite.

### 6.7.101 chart.dataZoom.selectedDataBackground.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.7.102 chart.dataZoom.selectedDataBackground.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.7.103 chart.dataZoom.selectedDataBackground.lineStyle.cap

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

#### 6.7.104 chart.dataZoom.selectedDataBackground.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.

- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

### Erlaubte Werte

- bevel
- round
- miter

### 6.7.105 chart.dataZoom.selectedDataBackground.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.7.106 chart.dataZoom.selectedDataBackground.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.7.107 chart.dataZoom.selectedDataBackground.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.7.108 chart.dataZoom.selectedDataBackground.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.109 chart.dataZoom.selectedDataBackground.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.110 chart.dataZoom.selectedDataBackground.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.111 chart.dataZoom.selectedDataBackground.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des selektierte Schattenbereichs.

### 6.7.112 chart.dataZoom.selectedDataBackground.areaStyle.color

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(47,69,84,0.3) ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.113 chart.dataZoom.selectedDataBackground.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.7.114 chart.dataZoom.selectedDataBackground.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 6.7.115 chart.dataZoom.selectedDataBackground.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.116 chart.dataZoom.selectedDataBackground.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.117 chart.dataZoom.selectedDataBackground.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.118 chart.dataZoom.moveHandleIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Symbol des Bewegungsgriiffs.

```
'M-320.9-50L-320.9-50c18.1,0,27.1,9,27.1,27.1V85.7c0,18.1-9,27.1-27.1,27.1l0,0c-18.1,0-27.1-9-27.1-27.1V-22.9C-348-41-339-50-320.9-50z
M-212.3-50L-212.3-50c18.1,0,27.1,9,27.1,27.1V85.7c0,18.1-9,27.1-27.1,27.1l0,0c-18.1,0-27.1-9-27.1-27.1V-22.9C-239.4-41-230.4-50-212.3-50z
M-103.7-50L-103.7-50c18.1,0,27.1,9,27.1,27.1V85.7c0,18.1-9,27.1-27.1,27.1l0,0c-18.1,0-27.1-9-27.1-27.1V-22.9C-130.9-41-121.8-50-103.7-50z'
```

Es kann mit 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild oder dataURI eines Bildes ist. Ein Beispiel für eine Bild-URL:

```
'image://http://xxx.xxx.xxx/a/b.png'
```

Ein dataURI-Beispiel:

```
'image://data:image/gif;base64,R0lGODlhEAAQAMQAORHHOVSKudfOulrSOp3W0yDZu6QdvCchPGolf00o/XBs/fNwfjZ0frl3/zy7///wAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACH5BAKAABAALAAAAAAQABAAAVVICSOZGLCQAsJ6mu7fiyZeKqNKToQGDsM8hBADgUXoGAiqhSvp5QAnQKGIgUhwFUYLCVDFCrKUE1lBaVAViFIDLTIbMC5Gm2hB0SlBCBMQiB0UjIA7'
```

Symbole können über 'path://' in ECharts auf einen beliebigen Vektorpfad gesetzt werden. Im Vergleich zu einem Rasterbild verhindern Vektorpfade beim Skalieren Zacken und Unschärfe und haben eine bessere Kontrolle über Farbänderungen. Die Größe des Vektorsymbols wird automatisch angepasst. Weitere Informationen zum Format des Pfads finden Sie unter SVG PathData. Sie können Vektorpfade aus Tools wie Adobe exportieren Zum Beispiel:

```
'path://
M30.9,53.2C16.8,53.2,5.3,41.7,5.3,27.6S16.8,2,30.9,2C45,2,56.4,13.5,56.4,27.6S45,53.2,30.9,53.2z
M30.9,3.5C17.6,3.5,6.8,14.4,6.8,27.6c0,13.3,10.8,24.1,24.101,24.1C44.2,51.7,55,40.9,55,27.6C54.9,14.4,44.1,3.5,30.9,3.5z
M36.9,35.8c0,0.601-0.4,1-0.9,1h-1.3c-0.5,0-0.9-0.399-0.9-1V19.5c0-0.6,0.4-1,0.9-1H36c0.5,0,0.9,0.4,0.9,1V35.8z M27.8,35.8c0,0.601-0.4,1-0.9,1h-1.3c-0.5,0-0.9-0.399-0.9-1V19.5c0-0.6,0.4-1,0.9-1'
```

### 6.7.119 chart.dataZoom.moveHandleSize

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default:7

Größe des Bewegungsgriffs.

### 6.7.120 chart.dataZoom.moveHandleStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des Bewegungsgriffs.

### 6.7.121 chart.dataZoom.moveHandleStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: #a7b7cc ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.7.122 chart.dataZoom.moveHandleStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.7.123 chart.dataZoom.moveHandleStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.7.124 chart.dataZoom.moveHandleStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.7.125 chart.dataZoom.moveHandleStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `borderType` können wir den Linienstil flexibler gestalten.

### 6.7.126 chart.dataZoom.moveHandleStyle.borderCap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : `butt`

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- `'butt'`: Die Linienenden werden an den Endpunkten abgerundet.
- `'round'`: Die Linienenden werden abgerundet.
- `'square'`: Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- `butt`
- `round`
- `square`

### 6.7.127 chart.dataZoom.moveHandleStyle.borderJoin

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.7.128 chart.dataZoom.moveHandleStyle.borderMiterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.7.129 chart.dataZoom.moveHandleStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.7.130 chart.dataZoom.moveHandleStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.7.131 chart.dataZoom.moveHandleStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.132 chart.dataZoom.moveHandleStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.133 chart.dataZoom.moveHandleStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.7.134 chart.dataZoom.brushSelect

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default: true

Ob der Pinsel zur Auswahl aktiviert werden soll.

### 6.7.135 chart.dataZoom.brushStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des Brushing Area.

### 6.7.136 chart.dataZoom.brushStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ default:rgba(135,175,274,0.15)]

Farbe.

### 6.7.137 chart.dataZoom.brushStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.7.138 chart.dataZoom.brushStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.7.139 chart.dataZoom.brushStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.7.140 chart.dataZoom.brushStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.7.141 chart.dataZoom.brushStyle.borderCap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.



## Erlaubte Werte

- butt
- round
- square

### 6.7.142 chart.dataZoom.brushStyle.borderJoin

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.7.143 chart.dataZoom.brushStyle.borderMiterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

### Schlagworte / Tags

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

## 6.7.144 chart.dataZoom.brushStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.7.145 chart.dataZoom.brushStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.7.146 chart.dataZoom.brushStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.7.147 chart.dataZoom.brushStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.7.148 chart.dataZoom.brushStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Data Zoom
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.8 ECharts (5) Gantt

### 6.8.1 chart.gantt.column.categoryname

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-category

Legt die Spaltennummer für den CategoryName fest (Informationen werden auf der linken Grüne-Tab angezeigt). Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.categoryname", "2");
```

### 6.8.2 chart.gantt.column.categorygroup

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-category

Legt die Spaltennummer für den CategoryGroup fest (Informationen werden neben der linken Grüne-Tab angezeigt). Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.categorygroup", "2");
```

### 6.8.3 chart.gantt.column.barinfo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt die Spaltennummer für die Informationen, die auf der Balken angezeigt wird. Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 0 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.barinfo", "2");
```

### 6.8.4 chart.gantt.column.iconinfo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die Spaltennummer für die Icon Informationen an den linken Seiten des Gantt-charts. Diese Spalte ist benutzt für die Entscheidung ob die Icon angezeigt werden soll oder nicht. Diese Option ist mit die folgende Option benutzt: "chart.gantt.iconlink.check".

Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 1 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.iconinfo", "2");
```

### 6.8.5 chart.gantt.column.barcolor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	bar-color, gantt

Legt die Spaltennummer für die Farbe der Balken. Hinweis: Die Spaltennummer dieser Spalte wird unter der Berücksichtigung angegeben, dass die Indizes im Gantt-Datensatz von 0 beginnen.

Beispiel:

```
dashlet.addOption("chart.gantt.column.barcolor", "2");
```

### 6.8.6 chart.gantt.screenitems

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-items

Legt fest, wie viele Datensätze im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.screenitems", "15");
```

### 6.8.7 chart.gantt.screenitems.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, gantt-items

Legt fest, die Höhe des Datensatzes im Gantt-Diagramm.

Beispiel:

```
dashlet.addOption("chart.gantt.screenitems.height", "15");
```

### 6.8.8 chart.gantt.iconlink.check

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt für die Entscheidung ob die Icon angezeigt werden soll oder nicht. Diese Option ist mit die folgende Option benutzt : "chart.gantt.column.iconinfo".

Beispiel:

```
dashlet.addOption("chart.gantt.iconlink.check", "'USA'");
```

## 6.8.9 chart.gantt.iconlink.path

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die SVG Image für das Icon fest.

Beispiel:

```
dashlet.addOption("chart.gantt.iconlink.path", "M369.9 97.9L286 14C277 5 264.8-.1
252.1-.1H48C21.5 0 0 21.5 0 48v416c0 26.5 21.5 48 48 48h288c26.5 0 48-21.5
48-48V131.9c0-12.7-5.1-25-14.1-34zM332.1 128H256V51.9l76.1 76.1zM48 464V48h160v104c0
13.3 10.7 24 24
24h104v288H48zm250.2-143.7c-12.2-12-47-8.7-64.4-6.5-17.2-10.5-28.7-25-36.8-46.3
3.9-16.1 10.1-40.6 5.4-56-4.2-26.2-37.8-23.6-42.6-5.9-4.4 16.1-.4 38.5 7 67.1-10
23.9-24.9 56-35.4 74.4-20 10.3-47 26.2-51 46.2-3.3 15.8 26 55.2 76.1-31.2 22.4-7.4
46.8-16.5 68.4-20.1 18.9 10.2 41 17 55.8 17 25.5 0 28-28.2 17.5-38.7zm-198.1
77.8c5.1-13.7 24.5-29.5 30.4-35-19 30.3-30.4 35.7-30.4 35zm81.6-190.6c7.4 0 6.7 32.1
1.8 40.8-4.4-13.9-4.3-40.8-1.8-40.8zm-24.4 136.6c9.7-16.9 18-37 24.7-54.7 8.3 15.1
18.9 27.2 30.1 35.5-20.8 4.3-38.9 13.1-54.8 19.2zm131.6-5s-5 6-37.3-7.8c35.1-2.6 40.9
5.4 37.3 7.8z");
```

## 6.8.10 chart.gantt.icon.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px;">TESTED</span>



<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, icon

Legt die Farbe für das Icon fest.

Beispiel:

```
dashlet.addOption("chart.gantt.icon.color", "#990000");
```

### 6.8.11 chart.gantt.nodata.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.text", "Keine Informationen");
```

### 6.8.12 chart.gantt.nodata.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text-Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.color","#0000ff");
```

### 6.8.13 chart.gantt.nodata.textColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Text-Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.textColor","#0000ff");
```

### 6.8.14 chart.gantt.grid.height

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt die Höhe des Grid fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.height" , "'auto'");
```

### 6.8.15 chart.gantt.grid.left

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die linke Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.left" , "100");
```

### 6.8.16 chart.gantt.grid.top

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die obere Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.top" , "100");
```

### 6.8.17 chart.gantt.grid.right

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die rechte Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.right" , "100");
```

### 6.8.18 chart.gantt.grid.bottom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt

Legt den Abstand für die untere Seite des Gantt-Diagramms fest

Beispiel:

```
dashlet.addOption("chart.gantt.grid.bottom" , "100");
```

### 6.8.19 chart.gantt.nodata.maskColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagnworte / Tags</b>	gantt, nodatainformation

Legt den anzuzeigenden Farbe fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.maskColor","#0000ff");
```

### 6.8.20 chart.gantt.nodata.zlevel

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagnworte / Tags</b>	gantt, nodatainformation

Legt die Z-Level für das Gantt-Diagramm.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.zlevel","0");
```

### 6.8.21 chart.gantt.nodata.fontSize

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0

<b>Schlagworte / Tags</b>	gantt, nodatainformation
---------------------------	--------------------------

Legt den anzuzeigenden Text-Size fest, wenn keine Informationen vorhanden sind, die im Gantt-Diagramm angezeigt werden sollen.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.fontSize","12");
```

## 6.8.22 chart.gantt.nodata.showSpinner

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Wenn true festgelegt ist, wird der Spinner angezeigt, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.showSpinner","true");
```

## 6.8.23 chart.gantt.nodata.spinnerRadius

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den Radius des Spinners fest, der angezeigt wird, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.spinnerRadius", "20");
```

### 6.8.24 chart.gantt.nodata.lineWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	gantt, nodatainformation

Legt den LineWidth des Spinners fest, der angezeigt wird, wenn im Gantt-Diagramm keine Informationen vorhanden sind.

Beispiel:

```
dashlet.addOption("chart.gantt.nodata.lineWidth", "2");
```

### 6.8.25 chart.gantt.axis.path

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die SVG Image für das Axes des Gantt-Charts fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.path", "M0,0 L0,-20 L30,-20 C42,-20 38,-1 50,-1 L70,-1 L70,0 Z");
```

### 6.8.26 chart.gantt.axis.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den X-Wert der linken oberen Ecke des Elements im Koordinatensystem der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.x", "0");
```

### 6.8.27 chart.gantt.axis.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den y-Wert der linken oberen Ecke des Elements im Koordinatensystem der Achse des Gantt-Diagramms fest.

Beispiel:



```
dashlet.addOption("chart.gantt.axis.y", "-20");
```

## 6.8.28 chart.gantt.axis.width

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Breite der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.width", "140");
```

## 6.8.29 chart.gantt.axis.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Höhe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.height", "40");
```

### 6.8.30 chart.gantt.axis.layout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt das Layout der Achse des Gantt-Diagramms fest. Mit dem Layout kann die Transformationsstrategie festgelegt werden.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.layout", "cover");
```

#### Erlaubte Werte

- cover
- center

### 6.8.31 chart.gantt.axis.style.fill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	

Die Farbe, die den Achsen-Pfad des Gantt-Diagramms ausfüllt.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.style.fill", "#368c6c");
```

### 6.8.32 chart.gantt.axis.categoryname.text.style.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den x-Wert der linken oberen Ecke des Elements im Koordinatensystem für den Kategorienamen der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categoryname.text.style.x", "4");
```

### 6.8.33 chart.gantt.axis.categoryname.text.style.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den y-Wert der linken oberen Ecke des Elements im Koordinatensystem für den Kategorienamen der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.y", "-3");
```

### 6.8.34 chart.gantt.axis.categorynametext.style.textVerticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die vertikale Textausrichtung für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textVerticalAlign", "bottom");
```

Erlaubte Werte

- top
- middle
- bottom

### 6.8.35 chart.gantt.axis.categorynametext.style.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2

<b>Schlagworte / Tags</b>	axis, gantt
---------------------------	-------------

Legt die horizontale Textausrichtung für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textAlign", "left");
```

### Erlaubte Werte

- left
- center
- right

### 6.8.36 chart.gantt.axis.categorynametext.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Textfarbe für den Kategorienamen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorynametext.style.textFill", "#fff");
```

### 6.8.37 chart.gantt.axis.categorygroupertext.style.x

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den X-Wert der linken oberen Ecke des Elements im Koordinatensystem für die Kategoriegruppe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.x", "125");
```

### 6.8.38 chart.gantt.axis.categorygroupertext.style.y

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt den Y-Wert der linken oberen Ecke des Elements im Koordinatensystem für die Kategoriegruppe der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.y", "-2");
```

### 6.8.39 chart.gantt.axis.categorygroupertext.style.textVerticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die vertikale Textausrichtung für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.textVerticalAlign",
"bottom");
```

Erlaubte Werte

- top
- middle
- bottom

#### 6.8.40 chart.gantt.axis.categorygroupertext.style.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die horizontale Textausrichtung für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.textAlign", "left");
```

Erlaubte Werte

- left
- center

- right

### 6.8.41 chart.gantt.axis.categorygroupertext.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Text-Farbe für die Kategoriegruppen in der Achse des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.axis.categorygroupertext.style.textFill", "#fff");
```

### 6.8.42 chart.gantt.rect.style.textFill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Text-Farbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.textFill", "#000");
```



### 6.8.43 chart.gantt.rect.style.stroke

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Stroke-Farbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.stroke", "transparent");
```

### 6.8.44 chart.gantt.rect.style.fill

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Gantt
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.2
<b>Schlagworte / Tags</b>	axis, gantt

Legt die Füllfarbe für die Balken des Gantt-Diagramms fest.

Beispiel:

```
dashlet.addOption("chart.gantt.rect.style.fill", "transparent");
```

## 6.9 ECharts (5) General

### 6.9.1 chart.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	map

Setzt der Typ des Charts.

Beispiel:

```
dashlet.addOption("chart.type", "bar");
```

### 6.9.2 chart.theme

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	theme

Legt die Thema für das Dashlet fest.

Beispiel:

```
<span style="color:black; font-size:10.0pt">dashlet.addOption</span>("chart.theme",  
infographics");
```

## 6.10 ECharts (5) Geo

### 6.10.1 chart.geo.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Komponente für geografische Koordinatensysteme.

Diese Komponente wird verwendet um Karten zu zeichnen, Streuserien und Balkenserien werden unterstützt.

Ab 3.1.10 unterstützt die geo-Komponente auch Maus-Events, deren Parameter sind:

```
{
  componentType: 'geo',
  // Index der geo-Komponente in den optionen
  geoIndex: number,
  // Name des geklickten Gebiets, z.B. Shanghai
  name: string,
  // Geklicktes Regionsobjekt als Input, siehe geo.regions
  region: Object
}
```

**Tip:** Die Regionsfarbe kann auch über Kartenserien gesteuert werden. Siehe series-map.geoIndex.

### 6.10.2 chart.geo.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.10.3 chart.geo.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die geo-Komponente angezeigt werden soll.

### 6.10.4 chart.geo.map

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Kartendiagramme.

Wegen der zunehmenden Feinheit von Karten enthält ECharts 3 aus Gründen der Paketgröße standardmäßig keine Kartendaten. Benötigte Kartendateien können auf der Karten-Downloadseite gefunden und dann in ECharts eingebunden und registriert werden.

ECharts stellt zwei Formate für Kartendaten zur Verfügung. Das eine kann in <script>-Tags als JavaScript-Datei eingebunden werden, das andere ist in JSON-Format und sollte mit AJAX geladen werden. Sobald die JavaScript-Datei geladen ist, werden Kartename und -daten automatisch geladen, während der Name in der JSON-Form explizit zugewiesen werden muss.

Hier sind Beispiele der beiden Typen:

#### JavaScript Import Beispiel

```
<script src="echarts.js"></script>
<script src="map/js/china.js"></script>
<script>
var chart = echarts.init(document.getElementById('main'));
chart.setOption({
  series: [{
    type: 'map',
    map: 'china'
  }]
});
</script>
```

### JSON Import Beispiel

```
$.get('map/json/china.json', function (chinaJson) {
  echarts.registerMap('china', chinaJson);
  var chart = echarts.init(document.getElementById('main'));
  chart.setOption({
    series: [{
      type: 'map',
      map: 'china'
    }]
  });
});
```

ECharts verwendet das geoJSON-Format für Kartenumrandungen. Neben den oben vorgestellten Methoden kann man, wenn man will, geoJSON-Daten auch mit anderen Methoden beschaffen und in ECharts registrieren.

### 6.10.5 chart.geo.roam

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Zoomen und Bewegen mit der Maus aktiviert werden soll, standardmäßig false. Wenn entweder zoomen oder bewegen gewünscht ist, kann der Wert auf 'scale' oder 'move' gesetzt werden. Andernfalls kann er auf true gesetzt werden, um beides zu aktivieren.

## Erlaubte Werte

- {boolean}
- scale
- move

### 6.10.6 chart.geo.center

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zentrum des momentanen Viewports, in Längen- und Breitengraden.

Beispiel:

```
center: [115.97, 29.71]
```

### 6.10.7 chart.geo.aspectScale

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.75 ]

Wird verwendet, um das Seitenverhältnis von geo zu skalieren.

Das endültige Seitenverhältnis berechnet sich aus:  $\text{geoBoundingBox.width} / \text{geoBoundingBox.height} * \text{aspectScale}$ .

### 6.10.8 chart.geo.boundingCoords

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zweidimensionales Array. Definiert die Koordinaten von oben-links und unten-rechts in der Layout-Box

```
// Eine komplette Weltkarte
map: 'world',
left: 0, top: 0, right: 0, bottom: 0,
boundingCoords: [
  // [Längengrad, Breitengrad] der Ecke oben-links
  [-180, 90],
  // [Längengrad, Breitengrad] der Ecke unten-rechts
  [180, -90]
],
```

### 6.10.9 chart.geo.zoom

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Zoom-Rate des momentanen Viewports.

### 6.10.10 chart.geo.scaleLimit.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grenzen der Skalierung, mit min und max. Standardmäßig 1.

### 6.10.11 chart.geo.scaleLimit.min

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimale Skalierung.

### 6.10.12 chart.geo.scaleLimit.max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Maximale Skalierung

### 6.10.13 chart.geo.nameMap

<b>Datentyp</b>	OBJECT
-----------------	--------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mapping der Namen für angepasste Gebiete. Zum Beispiel:

```
{
  'China' : '??'
}
```

#### 6.10.14 chart.geo.selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Entscheidung, ob mehrfache Auswahl unterstützt wird. Standardmäßig steht der Wert auf false um die Auswahl zu deaktivieren. Der Wert kann auch 'single' für das Auswählen einzelner Gebiete, oder 'multiple' für das Auswählen mehrerer Gebiete sein.

#### 6.10.15 chart.geo.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Textuelle Beschriftung der geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

## 6.10.16 chart.geo.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

## 6.10.17 chart.geo.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.10.18 chart.geo.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.10.19 chart.geo.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.10.20 chart.geo.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.10.21 chart.geo.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

**6.10.22 chart.geo.label.color**

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.23 chart.geo.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.10.24 chart.geo.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.10.25 chart.geo.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.10.26 chart.geo.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.10.27 chart.geo.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.10.28 chart.geo.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:



```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.10.29 chart.geo.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.10.30 chart.geo.label.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.31 chart.geo.label.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.32 chart.geo.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.10.33 chart.geo.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.10.34 chart.geo.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.10.35 chart.geo.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.10.36 chart.geo.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.10.37 chart.geo.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.10.38 chart.geo.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.10.39 chart.geo.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.10.40 chart.geo.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.10.41 chart.geo.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 6.10.42 chart.geo.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.10.43 chart.geo.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.10.44 chart.geo.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.10.45 chart.geo.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.10.46 chart.geo.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.10.47 chart.geo.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.10.48 chart.geo.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts

3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 6.10.49 chart.geo.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.10.50 chart.geo.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight

### 6.10.51 chart.geo.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.10.52 chart.geo.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.10.53 chart.geo.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.10.54 chart.geo.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.10.55 chart.geo.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.56 chart.geo.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.10.57 chart.geo.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.10.58 chart.geo.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.10.59 chart.geo.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.10.60 chart.geo.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

**Erlaubte Werte**

- left
- center
- right

**6.10.61 chart.geo.label.emphasis.verticalAlign**

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```



## Erlaubte Werte

- top
- middle
- bottom

### 6.10.62 chart.geo.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.10.63 chart.geo.label.emphasis.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.64 chart.geo.label.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.65 chart.geo.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.10.66 chart.geo.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.10.67 chart.geo.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.10.68 chart.geo.label.emphasis.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.10.69 chart.geo.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.10.70 chart.geo.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.10.71 chart.geo.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.10.72 chart.geo.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.10.73 chart.geo.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.10.74 chart.geo.label.emphasis.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 6.10.75 chart.geo.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.10.76 chart.geo.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.10.77 chart.geo.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.10.78 chart.geo.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.10.79 chart.geo.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.10.80 chart.geo.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.10.81 chart.geo.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grafikstil des Randes eines Kartengebiets, emphasis ist der Stil im hervorgehobenen Zustand, also wenn die Maus auf den Bereich zeigt oder er über die Legendenkomponente hervorgehoben wird.

## 6.10.82 chart.geo.itemStyle.areaColor.object

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Den Bereich füllende Farbe.

## 6.10.83 chart.geo.itemStyle.areaColor.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

#### 6.10.84 chart.geo.itemStyle.areaColor.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.10.85 chart.geo.itemStyle.areaColor.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.10.86 chart.geo.itemStyle.areaColor.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.10.87 chart.geo.itemStyle.areaColor.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.10.88 chart.geo.itemStyle.areaColor.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.10.89 chart.geo.itemStyle.areaColor.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.10.90 chart.geo.itemStyle.areaColor.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.10.91 chart.geo.itemStyle.areaColor.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.10.92 chart.geo.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil von Kartenbereich im hervorgehobenen Zustand.

### 6.10.93 chart.geo.itemStyle.emphasis.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Den Bereich füllende Farbe.

### 6.10.94 chart.geo.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.10.95 chart.geo.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.10.96 chart.geo.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.10.97 chart.geo.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.10.98 chart.geo.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.10.99 chart.geo.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.10.100 chart.geo.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.10.101 chart.geo.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.10.102 chart.geo.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.10.103 chart.geo.zlevel

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

#### 6.10.104 chart.geo.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

#### 6.10.105 chart.geo.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

#### 6.10.106 chart.geo.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle

- bottom

### 6.10.107 chart.geo.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 6.10.108 chart.geo.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Geo-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

## Erlaubte Werte

- {percent}
- {integer}

## 6.10.109 chart.geo.layoutCenter

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

layoutCenter und layoutSize stellen eine andere Layout-Strategie als left/right/top/bottom/width/height zur Verfügung.

Bei der Benutzung von left/right/top/bottom/width/height ist es schwierig, die Karte in einer Box mit festem Breite-Höhe-Verhältnis zu platzieren. In diesem Fall kann das layoutCenter-Attribut benutzt werden, um die Position des Zentrums der Karte zu definieren, und layoutSize kann benutzt werden, um die Größe der Karte zu definieren. Zum Beispiel:

```
layoutCenter: ['30%', '30%'],
// Ist das Breite-Höhe-Verhältnis größer als 1, wird die Breite auf 100 gesetzt.
// Sonst wird die Höhe auf 100 gesetzt.
// So wird sichergestellt, dass die Karten den 100x100 Bereich nicht überschreitet.
layoutSize: 100
```

Wurden diese beiden Werte gesetzt, so sind left/right/top/bottom/width/height ungültig.

## 6.10.110 chart.geo.layoutSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Karte, siehe layoutCenter für weitere Informationen. Prozentuale Werte relativ zur Bildschirmbreite und absolute Pixelwerte werden unterstützt.

### 6.10.111 chart.geo.regions.{x}.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil für bestimmte Regionen konfigurieren. Zum Beispiel:

```
regions: [{
  name: 'Guangdong',
  itemStyle: {
    areaColor: 'red',
    color: 'red'
  }
}]
```

Die Regionsfarbe kann auch über Kartenserien kontrolliert werden. Siehe series-map.geolIndex

### 6.10.112 chart.geo.regions.{x}.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name eines Gebiets auf der Karte, wie 'Guangdong' oder 'Zhejiang'.

### 6.10.113 chart.geo.regions.{x}.selected

<b>Datentyp</b>	BOOLEAN
-----------------	---------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob dieses Gebiet ausgewählt ist.

#### 6.10.114 chart.geo.regions.{x}.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil dieses Bereichs.

#### 6.10.115 chart.geo.regions.{x}.itemStyle.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsfarbe in der Karte

### 6.10.116 chart.geo.regions.{x}.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.10.117 chart.geo.regions.{x}.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.10.118 chart.geo.regions.{x}.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.10.119 chart.geo.regions.{x}.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.10.120 chart.geo.regions.{x}.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.10.121 chart.geo.regions.{x}.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.10.122 chart.geo.regions.{x}.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.10.123 chart.geo.regions.{x}.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.10.124 chart.geo.regions.{x}.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.10.125 chart.geo.regions.{x}.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil eines hervorgehobenen BereichsBereichs.

### 6.10.126 chart.geo.regions.{x}.itemStyle.emphasis.areaColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsfarbe in der Karte

### 6.10.127 chart.geo.regions.{x}.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.10.128 chart.geo.regions.{x}.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.10.129 chart.geo.regions.{x}.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.10.130 chart.geo.regions.{x}.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.10.131 chart.geo.regions.{x}.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.10.132 chart.geo.regions.{x}.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.10.133 chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 6.10.134 chart.geo.regions.{x}.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 6.10.135 chart.geo.regions.{x}.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.10.136 chart.geo.regions.{x}.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 6.10.137 chart.geo.regions.{x}.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.10.138 chart.geo.regions.{x}.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]

Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.10.139 chart.geo.regions.{x}.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.10.140 chart.geo.regions.{x}.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.10.141 chart.geo.regions.{x}.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.10.142 chart.geo.regions.{x}.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

**6.10.143 chart.geo.regions.{x}.label.color**

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.144 chart.geo.regions.{x}.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.10.145 chart.geo.regions.{x}.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'



- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bolder
- bold
- lighter
- {integer}

### 6.10.146 chart.geo.regions.{x}.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.10.147 chart.geo.regions.{x}.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.10.148 chart.geo.regions.{x}.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.10.149 chart.geo.regions.{x}.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.10.150 chart.geo.regions.{x}.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.10.151 chart.geo.regions.{x}.label.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.152 chart.geo.regions.{x}.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.153 chart.geo.regions.{x}.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.10.154 chart.geo.regions.{x}.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.10.155 chart.geo.regions.{x}.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.10.156 chart.geo.regions.{x}.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.10.157 chart.geo.regions.{x}.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.10.158 chart.geo.regions.{x}.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.10.159 chart.geo.regions.{x}.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.10.160 chart.geo.regions.{x}.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.10.161 chart.geo.regions.{x}.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.10.162 chart.geo.regions.{x}.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.163 chart.geo.regions.{x}.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------



<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.10.164 chart.geo.regions.{x}.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.10.165 chart.geo.regions.{x}.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.10.166 chart.geo.regions.{x}.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.10.167 chart.geo.regions.{x}.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.10.168 chart.geo.regions.{x}.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.10.169 chart.geo.regions.{x}.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen geo-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts

3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.

### 6.10.170 chart.geo.regions.{x}.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.10.171 chart.geo.regions.{x}.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte  
position: [10, 10],  
// Relative Prozentwerte  
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight

### 6.10.172 chart.geo.regions.{x}.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.10.173 chart.geo.regions.{x}.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.10.174 chart.geo.regions.{x}.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.10.175 chart.geo.regions.{x}.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird \n als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.10.176 chart.geo.regions.{x}.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.177 chart.geo.regions.{x}.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.10.178 chart.geo.regions.{x}.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}



## 6.10.179 chart.geo.regions.{x}.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.10.180 chart.geo.regions.{x}.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.10.181 chart.geo.regions.{x}.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

**Erlaubte Werte**

- left
- center
- right

**6.10.182 chart.geo.regions.{x}.label.emphasis.verticalAlign**

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 6.10.183 chart.geo.regions.{x}.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.10.184 chart.geo.regions.{x}.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.185 chart.geo.regions.{x}.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.186 chart.geo.regions.{x}.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.10.187 chart.geo.regions.{x}.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.10.188 chart.geo.regions.{x}.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.10.189 chart.geo.regions.{x}.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.10.190 chart.geo.regions.{x}.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.10.191 chart.geo.regions.{x}.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.10.192 chart.geo.regions.{x}.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.10.193 chart.geo.regions.{x}.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.10.194 chart.geo.regions.{x}.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Geo

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.10.195 chart.geo.regions.{x}.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.10.196 chart.geo.regions.{x}.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Rahmenbreite des Textes selbst.

### 6.10.197 chart.geo.regions.{x}.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.10.198 chart.geo.regions.{x}.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.10.199 chart.geo.regions.{x}.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.10.200 chart.geo.regions.{x}.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.10.201 chart.geo.regions.{x}.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.10.202 chart.geo.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Geo
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

## 6.11 ECharts (5) Graphic

### 6.11.1 chart.graphic.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Graphic
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Grafik-Komponente ermöglicht das Erstellen von grafischen Elementen in ECharts.

Die folgenden Grafiktypen werden unterstützt:

image, text, circle, sector, ring, polygon, polyline, rect, line, bezierCurve, arc, group

#### **Konfiguration der grafischen Komponente**

Eine einfache Art, um ein grafisches Element zu definieren:

```
myChart.setOption({
  ...,
  graphic: {
    type: 'image',
    ...
  }
});
```

Mehrere grafische Elemente definieren:

```
myChart.setOption({
  ...,
  graphic: [
    { // Ein 'image'-Element.
      type: 'image',
      ...
    },
    { // Ein 'text'-Element mit spezifizierter id.
      type: 'text',
      id: 'text1',
      ...
    },
    { // Ein 'group'-Element, in dem Kinder definiert werden können.
      type: 'group',
      children: [
        {
          type: 'rect',
          id: 'rect1',
          ...
        },
        {
          type: 'image',
          ...
        },
        ...
      ]
    }
  ],
  ...
});
```

Ersetzen und entfernen von Elementen mit setOption:

```
myChart.setOption({
  ...,
  graphic: [
    { // Das oben definierte Element 'text1' entfernen.
      id: 'text1',
      $action: 'remove',
      ...
    },
    { // Das oben definierte 'rect1' durch einen Zirkel ersetzen.
      // Man merke, obwohl 'rect1' ein Kind einer Gruppe ist,
      // ist es nicht nötig die Ebenenbeziehung erneut zu beachten,
      // wenn id benutzt wird, um das Element zu spezifizieren.
      id: 'rect1',
      $action: 'replace',
      type: 'circle',
      ...
    }
  ]
});
```

Es ist anzumerken, dass bei der Modifizierung von existierenden Elementen mit `setOption`, falls `id` nicht spezifiziert ist, die neuen Optionen auf die existierenden Elemente anhand ihrer Reihenfolge gemappt werden, was unerwartete Ergebnisse haben kann. Daher wird normalerweise die Verwendung von `id` empfohlen.

### Konfiguration von grafischen Elementen

Verschiedene Typen von grafischen Elementen haben ihre eigenen jeweiligen Konfigurationen, aber die folgenden Attribute haben sie gemein:

```
{
  // id wird verwendet, um das Element bei Updates zu spezifizieren.
  // id kann ignoriert werden, falls sie nicht benötigt wird.
  id: 'xxx',

  // Spezifiziert Elementtyp. Kann bei der ersten Definition eines Elements nicht
  ignoriert werden.
  type: 'image',

  // Alle folgenden Attribute können ignoriert werden und kriegen dann
  Standardwerte zugewiesen.

  // Spezifiziert die Operation, die auf dem Element ausgeführt werden soll, wenn
  `setOption` aufgerufen wird.
  // Standardwert ist 'merge', auch möglich ist 'replace' und 'remove'.
  $action: 'replace',

  // Diese vier Eigenschaften werden verwendet, um das Element zu platzieren.
  // Jeder Wert kann absolut (wie 10, was 10 Pixel bedeutet), relativ (wie '12%')
  oder 'center'/'middle' sein.
  left: 10,
  // right: 10,
  top: 'center',
  // bottom: '10%',

  shape: {
    // Hier sind Einstellungen für die Form, wie `x`, `y`, `cx`, `cy`, `width`,
    // `height`, `r`, `points`, ...
    // Man merke, falls `left`/`right`/`top`/`bottom` gesetzt wurden,
    funktionieren `x`/`y`/`cx`/`cy` hier nicht.
  },

  style: {
    // Hier sind Einstellungen für den Stil des Elements, wie `fill`, `stroke`,
    // `lineWidth`, `shadowBlur`, ...
  },

  // z-Wert der Elemente.
  z: 10,
  // Ob auf Maus-Events / Touch-Events reagiert werden soll.
  silent: true,
  // Ob das Element unsichtbar ist.
  invisible: false,
  // Spezifiziert, ob das gesamte transformierte Element (Kinder beinhaltend, falls
  es group ist)
  // durch seinen Container begrenzt wird. Mögliche Werte sind 'raw', 'all'.
  bounding: 'raw',
  // Kann gezogen werden oder nicht.
  draggable: false,
  // Event-handler, kann auch onmousemove, ondrag, ... sein (unten aufgeführt)
  onclick: function () {...}
}
```

## Event-handlers von grafischen Elementen

Diese Events werden unterstützt: onclick, onmouseout, onmouseover, onmousewheel, onmousedown, onmouseup, ondrag, ondragstart, ondragend, ondragcenter, ondragleave, ondragover, ondrop.

## Hierarchie von grafischen Elementen

Nur group-Elemente haben Kinder, wodurch eine Gruppe von Elementen zusammen positioniert und transformiert werden kann.

## Konfiguration der Form grafischer Elemente

Elemente mit verschiedenen Typen haben entsprechend unterschiedliche Einstellungen ihrer Form. Zum Beispiel:

```
{
  type: 'rect',
  shape: {
    x: 10,
    y: 10,
    width: 100,
    height: 200
  }
},
{
  type: 'circle',
  shape: {
    cx: 20,
    cy: 30,
    r: 100
  }
},
{
  type: 'image',
  style: {
    image: 'http://xxx.xxx.xxx/a.png',
    x: 100,
    y: 200,
    width: 230,
    height: 400
  }
},
{
  type: 'text',
  style: {
    text: 'Dieser text',
    x: 100,
    y: 200
  }
}
```

## Transformation und absolute Positionierung grafischer Elemente

Elemente können transformiert werden (Translation, Rotation, Skalierung). Siehe position, rotation, scale, origin

Zum Beispiel:



```
{
  type: 'rect', // oder beliebiger anderer Typ.

  // Translation, verwendet standardmäßig [0, 0].
  position: [100, 200],

  // Skalierung, verwendet standardmäßig [1, 1].
  scale: [2, 4],

  // Rotation, verwendet standardmäßig 0. Negative Werte stehen für Rotation im
  // Uhrzeigersinn.
  rotation: Math.PI / 4,

  // Ursprungspunkt der Rotation und Skalierung, verwendet standardmäßig [0, 0].
  origin: [10, 20],

  shape: {
    // ...
  }
}
```

- Jedes Element wird im Koordinatensystem seines Elternteils transformiert, insbesondere kann die Transformation von einem Element und seinem Elternteil "gestapelt" werden.
- Die Transformation wird in dieser Reihenfolge durchgeführt:
  - a. Translation  $[-el.origin[0], -el.origin[1]]$ .
  - b. Skalieren nach `el.scale`.
  - c. Rotation nach `el.rotation`.
  - d. Rücktranslation nach `el.origin`.
  - e. Translation nach `el.position`.
- Namentlich, Skalierung und Rotation zuerst, dann Translation. Durch diesen Vorgang beeinträchtigt die Translation nicht den Ursprung der Skalierung und Rotation.

### Relative Positionierung

In echten Anwendungen ist die Größe eines Containers nicht immer fest. Eine Möglichkeit zur relativen Positionierung wird also benötigt. In der Grafik-Komponente werden `left` / `right` / `top` / `bottom` / `width` / `height` verwendet, um Elemente relativ zu positionieren.

Zum Beispiel:

```

{ // Positioniere das Bild an der unteren Mitte des Containers.
  type: 'image',
  left: 'center', // An der horizontalen Mitte positionieren.
  bottom: '10%', // 10% über der unteren Grenze positionieren.
  style: {
    image: 'http://xxx.xxx.xxx/a.png',
    width: 45,
    height: 45
  }
},
{ // Die gesamte Gruppe in der rechten unteren Ecke des Containers positionieren.
  type: 'group',
  right: 0, // Positionierung an der rechten Grenze.
  bottom: 0, // Positionierung an der unteren Grenze.
  rotation: Math.PI / 4,
  children: [
    {
      type: 'rect',
      left: 'center', // Positionierung an der horizontalen Mitte nach dem
      Elternteil.
      top: 'middle', // Positionierung an der vertikalen Mitte nach dem
      Elternteil.
      shape: {
        width: 190,
        height: 90
      },
      style: {
        fill: '#fff',
        stroke: '#999',
        lineWidth: 2,
        shadowBlur: 8,
        shadowOffsetX: 3,
        shadowOffsetY: 3,
        shadowColor: 'rgba(0,0,0,0.3)'
      }
    },
    {
      type: 'text',
      left: 'center', // Positionierung an der horizontalen Mitte nach dem
      Elternteil.
      top: 'middle', // Positionierung an der vertikalen Mitte nach dem
      Elternteil.
      style: {
        fill: '#777',
        text: [
          'Dies ist Text',
          'Dies ist Text',
          'Text ausgeben'
        ].join('\n'),
        font: '14px Microsoft YaHei'
      }
    }
  ]
}

```

```
    ]
  }
```

Es ist anzumerken, dass `bounding` verwendet werden kann, um zu spezifizieren, ob das gesamte transformierte Element (Kinder beinhaltend, falls es `group` ist) durch seinen Container begrenzt wird.

## 6.12 ECharts (5) Grid

### 6.12.1 `chart.grid.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeichengitter in einem rechteckigen Koordinatensystem. In einem einzelnen Gitter sind maximal zwei X- und Y-Achsen erlaubt. Liniendiagramme, Balkendiagramme und Streudiagramme (Blasendiagramme) können in Gittern gezeichnet werden.

In ECharts 2.x konnte es nur maximal eine Gitterkomponente in einer einzelnen ECharts Instanz geben. In ECharts 3 gibt es jedoch keine solche Einschränkung.

### 6.12.2 `chart.grid.{x}.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.12.3 chart.grid.{x}.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Gitter im rechteckigen Koordinatensystem angezeigt werden soll.

### 6.12.4 chart.grid.{x}.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.12.5 chart.grid.{x}.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.12.6 chart.grid.{x}.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '10%' ]

Abstand zwischen der Gitterkomponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 6.12.7 chart.grid.{x}.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Gitter-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.12.8 chart.grid.{x}.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '10%' ]

Abstand zwischen der Gitterkomponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

#### Erlaubte Werte

- {integer}
- {percent}

### 6.12.9 chart.grid.{x}.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Gitterkomponente und der unteren Seite des Containers.

Platz für die Achsenbeschriftung und ggf. Legende.

Der Wert von bottom kann ein direkter Pixelwert wie '20px' sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Beispiel:

```
dashlet.addOption("chart.grid.bottom" , "250px");
```

#### Erlaubte Werte

- {integer}
- {percent}

### 6.12.10 chart.grid.{x}.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Gitterkomponente. Standardmäßig adaptiv.

### 6.12.11 chart.grid.{x}.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Gitterkomponente. Standardmäßig adaptiv.

Platz für die Achsenbeschriftung und ggf. Legende.

Beispiel:

```
dashlet.addOption("chart.grid.height" , "250px");
```

### 6.12.12 chart.grid.{x}.containLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Gitterregion die Achsentick-Beschriftungen der Achsen enthält.

- Wenn containLabel false ist:
  - grid.left grid.right grid.top grid.bottom grid.width grid.height bestimmen die Platzierung und Größe des Rechtecks, das aus xAxis und yAxis besteht.
  - Auf false setzen hilft, wenn mehrere Gitter anhand ihrer Achsen ausgerichtet werden müssen.
- Wenn containLabel true ist:
  - grid.left grid.right grid.top grid.bottom grid.width grid.height bestimmen die Platzierung und Größe des Rechtecks, das nicht nur die Achsen, sondern auch die Beschriftungen dieser Achsen enthält.
  - Auf true setzen hilft, wenn die länge der Achsenbeschriftungen dynamisch und schwer abzuschätzen ist, um zu verhindern, dass diese aus dem Container laufen oder andere Elemente überdecken



### 6.12.13 chart.grid.{x}.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Gitters, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 6.12.14 chart.grid.{x}.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Gitters. Unterstützt das gleiche Farbformat wie backgroundColor.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 6.12.15 chart.grid.{x}.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Gitters.

**Vorsicht:** Funktioniert nur wenn show auf true gesetzt ist.

### 6.12.16 chart.grid.{x}.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist und backgroundColor einen anderen Wert als transparent hat.

### 6.12.17 chart.grid.{x}.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 6.12.18 chart.grid.{x}.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 6.12.19 chart.grid.{x}.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Vorsicht:** Diese Eigenschaft funktioniert nur wenn show auf true gesetzt ist.

### 6.12.20 chart.grid.{x}.tooltip.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Koordinatensystems

### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

#### 6.12.21 chart.grid.{x}.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

#### 6.12.22 chart.grid.{x}.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentyphen in grid und polar unterstützt. Weiterhin lassen sich Achsen über `axisPointer.axis` zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item
- axis
- none

### 6.12.23 chart.grid.{x}.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

`tooltip.axisPointer` ist wie syntaktischer Zucker von `axisPointer`-Einstellungen an Achsen (zum Beispiel `xAxis.axisPointer` oder `angleAxis.axisPointer`). Detailliertere Einstellungen können an `someAxis.axisPointer` vorgenommen werden. Im Normalfall ist die Verwendung von `tooltip.axisPointer` jedoch praktischer.

**Hinweis:** Konfigurationen an `tooltip.axisPointer` haben eine niedrigere Priorität als `someAxis.axisPointer`

### 6.12.24 chart.grid.{x}.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

Erlaubte Werte

- line
- shadow
- none
- cross

### 6.12.25 chart.grid.{x}.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

Erlaubte Werte

- x
- y
- radius
- angle

- auto

### 6.12.26 chart.grid.{x}.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.12.27 chart.grid.{x}.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.12.28 chart.grid.{x}.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Beschriftung des axisPointers.

### 6.12.29 chart.grid.{x}.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.12.30 chart.grid.{x}.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 6.12.31 chart.grid.{x}.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

**Rückgabewert:**

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

## 6.12.32 chart.grid.{x}.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

## 6.12.33 chart.grid.{x}.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.12.34 chart.grid.{x}.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### 6.12.35 chart.grid.{x}.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.12.36 chart.grid.{x}.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.12.37 chart.grid.{x}.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.12.38 chart.grid.{x}.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.12.39 chart.grid.{x}.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.12.40 chart.grid.{x}.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.12.41 chart.grid.{x}.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.12.42 chart.grid.{x}.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.12.43 chart.grid.{x}.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.12.44 chart.grid.{x}.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.12.45 chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

#### 6.12.46 chart.grid.{x}.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

#### 6.12.47 chart.grid.{x}.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:



```

// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]

```

### 6.12.48 chart.grid.{x}.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.12.49 chart.grid.{x}.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.12.50 chart.grid.{x}.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.12.51 chart.grid.{x}.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.12.52 chart.grid.{x}.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.12.53 chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.12.54 chart.grid.{x}.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.12.55 chart.grid.{x}.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' gesetzt ist.

### 6.12.56 chart.grid.{x}.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.12.57 chart.grid.{x}.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.12.58 chart.grid.{x}.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.12.59 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.12.60 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.12.61 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.12.62 chart.grid.{x}.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.12.63 chart.grid.{x}.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.12.64 chart.grid.{x}.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.12.65 chart.grid.{x}.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.12.66 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.12.67 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.12.68 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.12.69 chart.grid.{x}.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.12.70 chart.grid.{x}.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.12.71 chart.grid.{x}.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

## 6.12.72 chart.grid.{x}.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.12.73 chart.grid.{x}.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.12.74 chart.grid.{x}.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.12.75 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.12.76 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.12.77 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.12.78 chart.grid.{x}.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.12.79 chart.grid.{x}.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.12.80 chart.grid.{x}.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 6.12.81 chart.grid.{x}.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}
```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 6.12.82 chart.grid.{x}.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 6.12.83 chart.grid.{x}.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 6.12.84 chart.grid.{x}.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 6.12.85 chart.grid.{x}.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.12.86 chart.grid.{x}.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 6.12.87 chart.grid.{x}.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.12.88 chart.grid.{x}.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.12.89 chart.grid.{x}.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.12.90 chart.grid.{x}.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.12.91 chart.grid.{x}.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.12.92 chart.grid.{x}.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.12.93 chart.grid.{x}.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.12.94 chart.grid.{x}.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.12.95 chart.grid.{x}.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.12.96 chart.grid.{x}.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.12.97 chart.grid.{x}.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.12.98 chart.grid.{x}.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.12.99 chart.grid.{x}.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.12.100 chart.grid.{x}.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.12.101 chart.grid.{x}.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 6.12.102 chart.grid.{x}.tooltip.axisPointer.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.12.103 chart.grid.{x}.tooltip.axisPointer.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate
- break
- breakAll

### 6.12.104 chart.grid.{x}.tooltip.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.12.105 chart.grid.{x}.tooltip.textStyle.overflow

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

Erlaubte Werte

- truncate
- break
- breakAll

### 6.12.106 chart.grid.{x}.tooltip.axisPointer.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.12.107 chart.grid.{x}.tooltip.axisPointer.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.12.108 chart.grid.{x}.tooltip.axisPointer.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.12.109 chart.grid.{x}.tooltip.axisPointer.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.



## Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

### 6.12.110 chart.grid.{x}.tooltip.axisPointer.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.12.111 chart.grid.{x}.tooltip.axisPointer.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 200 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.12.112 chart.grid.{x}.tooltip.axisPointer.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

### 6.12.113 chart.grid.{x}.tooltip.axisPointer.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Grid
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

## 6.13 ECharts (5) Legend

### 6.13.1 chart.legend.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Legendenkomponente.

Die Legendenkomponente zeigt die Symbole, Farben und Namen verschiedener Serien an. Man kann Legenden anklicken um das Anzeigen von Serien im Diagramm umzuschalten.

In EChart 3 kann eine einzelne ECharts-Instanz mehrere Legendenkomponenten enthalten, was das Layout solcher vereinfacht.

Gibt es zu viele Legendeneinträge, so sind vertikal oder horizontal scrollbare Legenden eine Option, um sie zu paginieren. Bitte legend.type betrachten.

### 6.13.2 chart.legend.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Typ der Legende. Optionale Werte:

- 'plain': Einfache Legende (Standard)
- 'scroll': Scrollbare Legende. Hilfreich wenn zu viele Legendeneinträge angezeigt werden müssen.

Wenn 'scroll' verwendet wird, können die untenstehenden Optionen für detailliertere Konfiguration verwendet werden:

- legend.scrollDataIndex
- legend.pageButtonItemGap
- legend.pageButtonGap
- legend.pageButtonPosition
- legend.pageFormatter
- legend.papelcons
- legend.papelconColor
- legend.papelconInactiveColor
- legend.papelconSize
- legend.pageTextStyle
- legend.animation
- legend.animationDurationUpdate

Erlaubte Werte

- plain
- scroll

### 6.13.3 chart.legend.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.13.4 chart.legend.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

### 6.13.5 chart.legend.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.13.6 chart.legend.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.13.7 chart.legend.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 6.13.8 chart.legend.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.13.9 chart.legend.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

### 6.13.10 chart.legend.bottom

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Legenden-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 6.13.11 chart.legend.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Legenden-Komponente. Standardmäßig adaptiv.

### 6.13.12 chart.legend.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Legenden-Komponente. Standardmäßig adaptiv.

### 6.13.13 chart.legend.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Layout-Orientierung der Legende.

#### Erlaubte Werte

- horizontal
- vertical

### 6.13.14 chart.legend.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Ausrichtung von Legenden-Marker und -Text. Wird standardmäßig automatisch aus Position und Orientierung der Komponente berechnet. Wenn der Wert von left dieser Komponente 'right' und das Layout vertikal ist (orient ist 'vertical'), so würde sie auf 'right' ausgerichtet werden.

## Erlaubte Werte

- auto
- left
- right

### 6.13.15 chart.legend.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der Legende um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.13.16 chart.legend.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Distanz zwischen den Legenden, horizontale Distanz im horizontalen Layout und vertikale Distanz im vertikalen layout.

### 6.13.17 chart.legend.itemWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 25 ]

Bildbreite des Legendensymbols.

### 6.13.18 chart.legend.itemHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Bildhöhe des Legendensymbols.

### 6.13.19 chart.legend.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob der Aspekt für Icons (von series.symbol oder nutzerdefiniertem legend.data.icon) in der Form von path:// behalten werden soll.

### 6.13.20 chart.legend.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatter wird verwendet, um das Label der Legende zu formatieren, welches String-Vorlagen und Callback-Funktionen unterstützt.

Beispiel:

```
// String-Vorlage benutzend, die Variable der Vorlage ist Legendename {name}
formatter: 'Legend {name}'
// Callback-Funktion benutzend
formatter: function (name) {
  return 'Legend ' + name;
}
```

### 6.13.21 chart.legend.selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ausgewählter Legendenmodus, welcher kontrolliert, ob das Anzeigen von Serien durch Klicken ihrer Legenden umgeschaltet werden kann. Dies ist standardmäßig möglich und durch setzen dieses Wertes auf false wird es verhindert.

Außerdem kann dieser Wert auf 'single' oder 'multiple' gesetzt werden, für Einzelauswahl oder Mehrfachauswahl.

### 6.13.22 chart.legend.inactiveColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Farbe der Legende, wenn diese nicht ausgewählt ist.

### 6.13.23 chart.legend.selected

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Statustabelle der ausgewählten Legenden.

Beispiel:

```
selected: {
  // ausgewählte 'series 1'
  'series 1': true,
  // nicht ausgewählte 'series 2'
  'series 2': false
}
```

### 6.13.24 chart.legend.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Legendentexts.

### 6.13.25 chart.legend.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.13.26 chart.legend.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 6.13.27 chart.legend.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 6.13.28 chart.legend.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.13.29 chart.legend.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.13.30 chart.legend.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.13.31 chart.legend.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.13.32 chart.legend.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.13.33 chart.legend.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.13.34 chart.legend.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.13.35 chart.legend.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.13.36 chart.legend.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[Default-Wert: 0]

Rahmenradius des Textfragments.

### 6.13.37 chart.legend.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.13.38 chart.legend.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.13.39 chart.legend.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.13.40 chart.legend.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.13.41 chart.legend.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.13.42 chart.legend.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## Erlaubte Werte

- {percent}

## 6.13.43 chart.legend.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.13.44 chart.legend.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.13.45 chart.legend.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.13.46 chart.legend.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.13.47 chart.legend.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.13.48 chart.legend.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.13.49 chart.legend.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.13.50 chart.legend.textStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.13.51 chart.legend.tooltip

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfiguration für den Tooltip der Legende.

### 6.13.52 chart.legend.data.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Datenarray der Legende. Ein Element des Arrays ist normalerweise ein String, der den Namen einer Serie (oder, in einem Kuchendiagramm, den Namen eines einzelnen Datenwerts) repräsentiert. Die Legendenkomponente berechnet dann die zugehörige Farbe und das Icon automatisch anhand der Serie. Die Spezialstrings "" (null-String) oder '\n' (Zeilenumbruch-String) können benutzt werden, um eine neue Zeile zu erhalten.

Wird data nicht spezifiziert, so werden die Namen automatisch aus den Serien geholt. Bei den meisten Serien werden diese aus series.name oder dem Dimensionsnamen, der durch seriesName von series.encode spezifiziert wird, bezogen. Bei manchen Serientypen, zum Beispiel bei zu Kuchen- oder Trichterdiagrammen gehörigen, kommen sie aus dem Namensfeld von series.data.

Soll der Stil eines Elementes festgelegt werden, kann diese Konfiguration mit angegeben werden. In diesem Fall wird ein name-Attribut verwendet um den Namen der Serie anzugeben.

Beispiel:

```
data: [{
  name: 'series 1',
  // erzwinge als Icon einen Kreis
  icon: 'circle',
  // setze die Textfarbe auf Rot
  textStyle: {
    color: 'red'
  }
}]
```

### 6.13.53 chart.legend.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe der Legende, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

## 6.13.54 chart.legend.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe der Legende. Unterstützt das gleiche Farbformat wie backgroundColor.

## 6.13.55 chart.legend.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite der Legende.

### 6.13.56 chart.legend.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 6.13.57 chart.legend.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 6.13.58 chart.legend.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.13.59 chart.legend.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.13.60 chart.legend.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

### 6.13.61 chart.legend.scrollDataIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

dataIndex des linksobert angezeigten Eintrags.

Obwohl das Scrollen von Legendeneinträgen kontrolliert werden kann indem setOption aufgerufen wird und diese Eigenschaft gesetzt wird, wird empfohlen, dies nicht zu tun, falls es nicht nötig ist (setOption kann Zeitaufwändig sein) und stattdessen die Aktion legendScroll zu verwenden.

### 6.13.62 chart.legend.pageButtonItemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Abstand zwischen den Seitenknöpfen und dem Seiteninfotext.

### 6.13.63 chart.legend.pageButtonGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Abstand zwischen den Seitenknöpfen und den Legendeneinträgen.

### 6.13.64 chart.legend.pageButtonPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'end' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Position der Seitenknöpfe und der Seiteninfo.

Mögliche Werte:

- 'start': Links oder oben.
- 'end': Rechts oder unten

### 6.13.65 chart.legend.pageFormatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: '{current}/{total}' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Formatierer der Seiteninfo. Steht standardmäßig auf '{current}/{total}', wobei {current} die aktuelle Seitennummer (mit 1 anfangend), und {total} die Gesamtzahl an Seiten ist.

Ist pageFormatter eine Funktion, so sollte sie einen String zurückgeben. Die Parameter sind:

```
{
  current: number
  total: number
}
```

### 6.13.66 chart.legend.pageIcons

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Icons der Seitenknöpfe.

### 6.13.67 chart.legend.pageIcons.horizontal

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Icons der Seitenknöpfe wenn legend.orient auf 'horizontal' steht.

Der Wert sollte ein Array, [Knopf für vorherige Seite, Knopf für nächste Seite] sein, standardmäßig ['M0,0L12,-10L12,10z', 'M0,0L-12,-10L-12,10z'].

Für jedes Element des Arrays:

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.13.68 chart.legend.pageIcons.vertical

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Icons der Seitenknöpfe wenn legend.orient auf 'vertical' steht.

Der Wert sollte ein Array, [Knopf für vorherige Seite, Knopf für nächste Seite] sein, standardmäßig ['M0,0L20,0L10,-20z', 'M0,0L20,0L10,20z'].

Für jedes Element des Arrays:

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.13.69 chart.legend.pageIconColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#2f4554' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Farbe der Seitenknöpfe.

### 6.13.70 chart.legend.pageIconInactiveColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#aaa' ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Farbe der Seitenknöpfe, wenn sie inaktiv sind.

### 6.13.71 chart.legend.pageIconSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Die Größe der Seitenknöpfe. Kann eine Zahl sein oder ein Array wie [10, 3], welches [Breite, Höhe] repräsentiert.

### 6.13.72 chart.legend.pageTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktioniert wenn legend.type auf 'scroll' gestellt ist.

Der Textstil der Seiteninfo.

### 6.13.73 chart.legend.pageTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

Textfarbe.

### 6.13.74 chart.legend.pageTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'

- 'oblique'

### Erlaubte Werte

- normal
- italic
- oblique

### 6.13.75 chart.legend.pageTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 6.13.76 chart.legend.pageTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.13.77 chart.legend.pageTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.13.78 chart.legend.pageTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.13.79 chart.legend.pageTextStyle.width

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.13.80 chart.legend.pageTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.13.81 chart.legend.pageTextStyle.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.13.82 chart.legend.pageTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.13.83 chart.legend.pageTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.



### 6.13.84 chart.legend.pageTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.13.85 chart.legend.pageTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.13.86 chart.legend.pageTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.13.87 chart.legend.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob das Scrollen animiert sein soll.

### 6.13.88 chart.legend.animationDurationUpdate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 800 ]

Dauer der Animation des Scrollens.

### 6.13.89 chart.legend.textStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

### Schlagworte / Tags

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

### Erlaubte Werte

- truncate
- break
- breakAll

## 6.13.90 chart.legend.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

## 6.13.91 chart.legend.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Symbol der Legendenelemente. Von ECharts bereitgestellte Symboltypen umfassen 'Kreis', 'Rect', 'RoundRect', 'Triangle', 'Diamond', 'Pin', 'Pfeil', 'None' Es kann mit 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild oder dataURI eines Bildes ist.

### 6.13.92 chart.legend.pageTextStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate
- break
- breakAll

### 6.13.93 chart.legend.pageTextStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.13.94 chart.legend.selector

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die Auswahlschaltfläche in der Legendenkomponente. Enthält derzeit sowohl eine vollständige Auswahl als auch eine inverse Auswahl. Die Auswahlschaltfläche wird standardmäßig nicht angezeigt, der Benutzer kann sie manuell konfigurieren.

Verwendungszweck:

```

selector: [
  {
    type: 'all or inverse',
    // can be any title you like
    title: 'All'
  },
  {
    type: 'inverse',
    title: 'Inv'
  }
]

// or
selector: true

// or
selector: ['all', 'inverse']

```

### 6.13.95 chart.legend.selectorPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : auto

Die Position der Auswahlschaltfläche, die am Ende oder Anfang der Legendenkomponente platziert werden kann, die entsprechenden Werte sind „end“ und „start“. Wenn die Legende horizontal angeordnet ist, wird der Selektor

standardmäßig an deren Ende platziert, und wenn die Legende vertikal angeordnet wird, wird der Selektor an deren Anfang platziert.

## Erlaubte Werte

- end
- start
- auto

### 6.13.96 chart.legend.selectorItemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die Lücke zwischen der Auswahltaste.

### 6.13.97 chart.legend.selectorButtonGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die Lücke zwischen Auswahl Schaltfläche und Legendenkomponente.

### 6.13.98 chart.legend.selectorLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der Textbeschriftungsstil der Auswahl Schaltfläche, der standardmäßig angezeigt wird.

### 6.13.99 chart.legend.selectorLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Gibt an ob die Label angezeigt wird.

### 6.13.100 chart.legend.selectorLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum Host-Grafikelement.

### 6.13.101 chart.legend.selectorLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.13.102 chart.legend.selectorLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.13.103 chart.legend.selectorLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

Wenn sie auf „inherit“ eingestellt ist, wird die Farbe als visuelle Farbe zugewiesen, z. B. als Serienfarbe.



### 6.13.104 chart.legend.selectorLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.13.105 chart.legend.selectorLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.13.106 chart.legend.selectorLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.13.107 chart.legend.selectorLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.13.108 chart.legend.selectorLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.13.109 chart.legend.selectorLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

### Erlaubte Werte

- top
- middle
- bottom

### 6.13.110 chart.legend.selectorLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.13.111 chart.legend.selectorLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wenn sie auf „inherit“ eingestellt ist, wird die Farbe als visuelle Farbe zugewiesen, z. B. als Serienfarbe.

### 6.13.112 chart.legend.selectorLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wenn sie auf „inherit“ eingestellt ist, wird die Farbe als visuelle Farbe zugewiesen, z. B. als Serienfarbe.

### 6.13.113 chart.legend.selectorLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.13.114 chart.legend.selectorLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.13.115 chart.legend.selectorLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.13.116 chart.legend.selectorLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.13.117 chart.legend.selectorLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.13.118 chart.legend.selectorLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.13.119 chart.legend.selectorLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.13.120 chart.legend.selectorLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Breite des Textblocks.

### 6.13.121 chart.legend.selectorLabel.height

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Höhe des Textblocks.

### 6.13.122 chart.legend.selectorLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wenn sie auf „inherit“ eingestellt ist, wird die Farbe als visuelle Farbe zugewiesen, z. B. als Serienfarbe.

### 6.13.123 chart.legend.selectorLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.13.124 chart.legend.selectorLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.13.125 chart.legend.selectorLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.13.126 chart.legend.selectorLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.13.127 chart.legend.selectorLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.13.128 chart.legend.selectorLabel.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

Erlaubte Werte

- truncate
- break
- breakAll

### 6.13.129 chart.legend.selectorLabel.ellipsis

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.13.130 chart.legend.selectorLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Legend
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

## 6.14 ECharts (5) Parallel

### 6.14.1 chart.parallel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

#### Einführung über parallele Koordinaten

Parallele Koordinaten sind ein häufig genutzter Weg um hochdimensionale Geometrie zu visualisieren und multivariate Daten zu analysieren.

Zum Beispiel, sei `series-parallel.data` die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
    lineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

Parallele Koordinaten werden oft verwendet um Mehrdimensionale Daten wie oben zusammenzufassen. Jede Achse steht für eine Dimension (also eine Spalte) und jede Linie steht für ein Datenelement. Daten können auf den Achsen mit dem brush ausgewählt werden.

### Kurzfassung der Konfiguration

Die grundlegende Konfiguration von parallelen Koordinaten läuft wie folgt ab:

```

option = {
  parallelAxis: [                                     // Definition der Achsen.
    {dim: 0, name: schema[0].text}, // Jede Achse hat ein 'dim'-Attribut, welches
    // für den Index der Dimension in den Daten steht.
    {dim: 1, name: schema[1].text},
    {dim: 2, name: schema[2].text},
    {dim: 3, name: schema[3].text},
    {dim: 4, name: schema[4].text},
    {dim: 5, name: schema[5].text},
    {dim: 6, name: schema[6].text},
    {dim: 7, name: schema[7].text,
      type: 'category', // Unterstützt auch Kategoriedaten.
      data: ['Exzellent', 'gut', 'leichte Verschmutzung', 'moderate
Verschmutzung', 'schwere Verschmutzung', 'massive Verschmutzung']
    }
  ],
  parallel: {                                         // Definition eines parallelen
  Koordinatensystems.
    left: '5%',                                       // Platzierung des parallelen
  Koordinatensystems.
    right: '13%',
    bottom: '10%',
    top: '20%',
    parallelAxisDefault: { // Ein Muster für die Achsendefinition, damit
  dieses nicht in 'parallelAxis' wiederholt werden muss.
      type: 'value',
      nameLocation: 'end',
      nameGap: 20
    }
  },
  series: [                                           // Die drei Serien, die sich das parallele
  Kordinatensystem teilen.
    {
      name: 'Beijing',
      type: 'parallel', // Der Typ dieser Serie ist 'parallel'.
      data: [
        [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
        [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Shanghai',
      type: 'parallel',
      data: [
        [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
        [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Guangzhou',

```

```

        type: 'parallel',
        data: [
            [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
            [5, 42, 24, 44, 0.76, 40, 16, 'exzellent'],
            ...
        ]
    }
]
};

```

Drei Komponenten sind hier involviert: `parallel`, `parallelAxis`, `series-parallel`

- `parallel`  
Diese Komponente ist das Koordinatensystem. Eine oder mehrere Serien (wie "Beijing", "Shanghai" und "Guangzhou" im obigen Beispiel) können sich ein Koordinatensystem teilen. Wie andere Koordinatensysteme können mehrere parallele Koordinatensysteme in einer ECharts-Instanz generiert werden. Das Setzen der Position wird auch hier durchgeführt.
- `parallelAxis`  
Dies ist die Achsenkonfiguration. Mehrere Achsen sind für parallele Koordinatensysteme notwendig. `parallelAxis.parallelIndex` wird verwendet um zu spezifizieren, zu welchem Koordinatensystem die Achse gehört. Standardmäßig wird das erste Koordinatensystem verwendet.
- `series-parallel`  
Dies ist die Definition der parallelen Serien, welche im parallelen Koordinatensystem gezeichnet werden.

### Anmerkungen und Best Practices

Wenn mehrere `parallelAxis` konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter `parallel.parallelAxisDefault` definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von `parallel.parallelAxisDefault` in `parallelAxis` eingefügt, um die finale Achesnkonfiguration zu erhalten.

### Falls die Daten zu groß sind und für schlechte Performanz sorgen

Es wird empfohlen, `series-parallel.lineStyle.width` auf 0.5 (oder weniger) zu setzen, was die Performanz signifikant erhöhen kann.

### Darstellung hochdimensionaler Daten

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann `parallel.axisExpandable` verwendet werden, um die Darstellung zu verbessern.

## 6.14.2 `chart.parallel.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 6.14.3 chart.parallel.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

## 6.14.4 chart.parallel.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.14.5 chart.parallel.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 6.14.6 chart.parallel.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

## 6.14.7 chart.parallel.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 80 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der rechten Seite des Containers.

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein.

### Erlaubte Werte

- {integer}
- {percent}

## 6.14.8 chart.parallel.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 60 ]

Abstand zwischen der Komponente des parallelen Koordinatensystems und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

### 6.14.9 chart.parallel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Komponente des parallelen Koordinatensystems. Standardmäßig adaptiv.

### 6.14.10 chart.parallel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Komponente des parallelen Koordinatensystems. Standardmäßig adaptiv.

### 6.14.11 chart.parallel.layout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Layout-Modi, deren optionale Werte sind:

- 'horizontal': Platziere jede Achse horizontal.
- 'vertical': Platziere jede Achse vertikal.

Erlaubte Werte

- horizontal
- vertical

### 6.14.12 chart.parallel.axisExpandable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann parallel.axisExpandable verwendet werden, um die Darstellung zu verbessern.

### 6.14.13 chart.parallel.axisExpandCenter

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Index der Achse, die zu Beginn als Zentrum der Expansion verwendet wird. Hat keinen standard-Wert und muss manuell zugewiesen werden.

Für weitere Informationen bitte an `parallel.axisExpandable` wenden.

### 6.14.14 `chart.parallel.axisExpandCount`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Definiert, wie viele Achsen zu Beginn im expandierten Zustand sind. Es wird empfohlen, diesen Wert manuell den Dimensionen entsprechend zu setzen.

Für weitere Informationen bitte an `parallel.axisExpandCenter` und `parallel.axisExpandable` wenden.

### 6.14.15 `chart.parallel.axisExpandWidth`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand in Pixeln zwischen zwei Achsen im expandierten Zustand.

Für weitere Informationen bitte an `parallel.axisExpandable` wenden.

### 6.14.16 chart.parallel.axisExpandTriggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'click' ]

Mögliche Werte:

- 'click': Löse Expansion durch Mausklicks aus.
- 'mousemove': Löse Expansion durch Mausbewegungen aus.

Erlaubte Werte

- click
- mousemove

### 6.14.17 chart.parallel.parallelAxisDefault.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn mehrere parallelAxis konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter parallel.parallelAxisDefault definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von parallel.parallelAxisDefault in parallelAxis eingefügt, um die finale Achsenkonfiguration zu erhalten.

### 6.14.18 chart.parallel.parallelAxisDefault.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entschieden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 6.14.19 chart.parallel.parallelAxisDefault.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

### 6.14.20 chart.parallel.parallelAxisDefault.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

#### 6.14.21 chart.parallel.parallelAxisDefault.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

#### 6.14.22 chart.parallel.parallelAxisDefault.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens, verwendet standardmäßig axisLine.lineStyle.color

### 6.14.23 chart.parallel.parallelAxisDefault.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.14.24 chart.parallel.parallelAxisDefault.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

#### Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

#### Erlaubte Werte

- normal
- bold

- bolder
- lighter
- {integer}

### 6.14.25 chart.parallel.parallelAxisDefault.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.14.26 chart.parallel.parallelAxisDefault.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 6.14.27 chart.parallel.parallelAxisDefault.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

#### 6.14.28 chart.parallel.parallelAxisDefault.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.14.29 chart.parallel.parallelAxisDefault.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.14.30 chart.parallel.parallelAxisDefault.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.14.31 chart.parallel.parallelAxisDefault.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.14.32 chart.parallel.parallelAxisDefault.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.14.33 chart.parallel.parallelAxisDefault.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.14.34 chart.parallel.parallelAxisDefault.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.14.35 chart.parallel.parallelAxisDefault.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.14.36 chart.parallel.parallelAxisDefault.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.14.37 chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.14.38 chart.parallel.parallelAxisDefault.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

## 6.14.39 chart.parallel.parallelAxisDefault.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.14.40 chart.parallel.parallelAxisDefault.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.14.41 chart.parallel.parallelAxisDefault.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

#### 6.14.42 chart.parallel.parallelAxisDefault.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 6.14.43 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 6.14.44 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 6.14.45 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

#### 6.14.46 chart.parallel.parallelAxisDefault.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

#### 6.14.47 chart.parallel.parallelAxisDefault.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

#### 6.14.48 chart.parallel.parallelAxisDefault.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 6.14.49 chart.parallel.parallelAxisDefault.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 6.14.50 chart.parallel.parallelAxisDefault.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 6.14.51 chart.parallel.parallelAxisDefault.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 6.14.52 chart.parallel.parallelAxisDefault.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.14.53 chart.parallel.parallelAxisDefault.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 6.14.54 chart.parallel.parallelAxisDefault.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

## 6.14.55 chart.parallel.parallelAxisDefault.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]



Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.14.56 chart.parallel.parallelAxisDefault.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.14.57 chart.parallel.parallelAxisDefault.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.14.58 chart.parallel.parallelAxisDefault.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.14.59 chart.parallel.parallelAxisDefault.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 6.14.60 chart.parallel.parallelAxisDefault.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

#### 6.14.61 chart.parallel.parallelAxisDefault.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

#### 6.14.62 chart.parallel.parallelAxisDefault.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.14.63 chart.parallel.parallelAxisDefault.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.14.64 chart.parallel.parallelAxisDefault.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.14.65 chart.parallel.parallelAxisDefault.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.14.66 chart.parallel.parallelAxisDefault.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.14.67 chart.parallel.parallelAxisDefault.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

## Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.14.68 chart.parallel.parallelAxisDefault.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.14.69 chart.parallel.parallelAxisDefault.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.14.70 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.14.71 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 6.14.72 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 6.14.73 chart.parallel.parallelAxisDefault.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 6.14.74 chart.parallel.parallelAxisDefault.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.14.75 chart.parallel.parallelAxisDefault.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.14.76 chart.parallel.parallelAxisDefault.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.14.77 chart.parallel.parallelAxisDefault.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.14.78 chart.parallel.parallelAxisDefault.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.14.79 chart.parallel.parallelAxisDefault.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.14.80 chart.parallel.parallelAxisDefault.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.14.81 chart.parallel.parallelAxisDefault.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Stil der Achsenlinie.

### 6.14.82 chart.parallel.parallelAxisDefault.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Farbe der Achsenticks wird standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.14.83 chart.parallel.parallelAxisDefault.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.14.84 chart.parallel.parallelAxisDefault.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.14.85 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.14.86 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.14.87 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.14.88 chart.parallel.parallelAxisDefault.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.14.89 chart.parallel.parallelAxisDefault.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.14.90 chart.parallel.parallelAxisDefault.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

#### 6.14.91 chart.parallel.parallelAxisDefault.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

#### 6.14.92 chart.parallel.parallelAxisDefault.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.14.93 chart.parallel.parallelAxisDefault.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.14.94 chart.parallel.parallelAxisDefault.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.



### 6.14.95 chart.parallel.parallelAxisDefault.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 6.14.96 chart.parallel.parallelAxisDefault.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```

// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getFullYear());
  }
  return texts.join('/');
}

```

### 6.14.97 chart.parallel.parallelAxisDefault.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.14.98 chart.parallel.parallelAxisDefault.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.14.99 chart.parallel.parallelAxisDefault.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.14.100 chart.parallel.parallelAxisDefault.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.14.101 chart.parallel.parallelAxisDefault.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.14.102 chart.parallel.parallelAxisDefault.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.14.103 chart.parallel.parallelAxisDefault.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.14.104 chart.parallel.parallelAxisDefault.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 6.14.105 chart.parallel.parallelAxisDefault.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.14.106 chart.parallel.parallelAxisDefault.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.14.107 chart.parallel.parallelAxisDefault.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

#### 6.14.108 chart.parallel.parallelAxisDefault.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

#### 6.14.109 chart.parallel.parallelAxisDefault.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.



### 6.14.110 chart.parallel.parallelAxisDefault.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.14.111 chart.parallel.parallelAxisDefault.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.14.112 chart.parallel.parallelAxisDefault.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.14.113 chart.parallel.parallelAxisDefault.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.14.114 chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.14.115 chart.parallel.parallelAxisDefault.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.14.116 chart.parallel.parallelAxisDefault.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.14.117 chart.parallel.parallelAxisDefault.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.14.118 chart.parallel.parallelAxisDefault.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.14.119 chart.parallel.parallelAxisDefault.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.14.120 chart.parallel.parallelAxisDefault.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 6.14.121 chart.parallel.parallelAxisDefault.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 6.14.122 chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.14.123 chart.parallel.parallelAxisDefault.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.14.124 chart.parallel.parallelAxisDefault.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

## 6.15 ECharts (5) Parallel Axis

### 6.15.1 chart.parallelAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Komponente ist die Koordinatenachse für ein paralleles Koordinatensystem.

## Einführung über parallele Koordinaten

Parallele Koordinaten sind ein häufig genutzter Weg um hochdimensionale Geometrie zu visualisieren und multivariate Daten zu analysieren.

Zum Beispiel, sei `series-parallel.data` die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
    lineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

Parallele Koordinaten werden oft verwendet um Mehrdimensionale Daten wie oben zusammenzufassen. Jede Achse steht für eine Dimension (also eine Spalte) und jede Linie steht für ein Datenelement. Daten können auf den Achsen mit dem brush ausgewählt werden.

## Kurzfassung der Konfiguration

Die grundlegende Konfiguration von parallelen Koordinaten läuft wie folgt ab:



```

option = {
  parallelAxis: [                                     // Definition der Achsen.
    {dim: 0, name: schema[0].text}, // Jede Achse hat ein 'dim'-Attribut, welches
    // für den Index der Dimension in den Daten steht.
    {dim: 1, name: schema[1].text},
    {dim: 2, name: schema[2].text},
    {dim: 3, name: schema[3].text},
    {dim: 4, name: schema[4].text},
    {dim: 5, name: schema[5].text},
    {dim: 6, name: schema[6].text},
    {dim: 7, name: schema[7].text,
      type: 'category', // Unterstützt auch Kategoriedaten.
      data: ['Exzellent', 'gut', 'leichte Verschmutzung', 'moderate
Verschmutzung', 'schwere Verschmutzung', 'massive Verschmutzung']
    }
  ],
  parallel: {                                       // Definition eines parallelen
    // Koordinatensystems.
    left: '5%', // Platzierung des parallelen
    // Koordinatensystems.
    right: '13%',
    bottom: '10%',
    top: '20%',
    parallelAxisDefault: { // Ein Muster für die Achsendefinition, damit
    // dieses nicht in 'parallelAxis' wiederholt werden muss.
      type: 'value',
      nameLocation: 'end',
      nameGap: 20
    }
  },
  series: [ // Die drei Serien, die sich das parallele
    // Koordinatensystem teilen.
    {
      name: 'Beijing',
      type: 'parallel', // Der Typ dieser Serie ist 'parallel'.
      data: [
        [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
        [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Shanghai',
      type: 'parallel',
      data: [
        [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
        [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
        ...
      ]
    },
    {
      name: 'Guangzhou',

```

```

        type: 'parallel',
        data: [
            [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
            [5, 42, 24, 44, 0.76, 40, 16, 'exzellent'],
            ...
        ]
    }
]
};

```

Drei Komponenten sind hier involviert: `parallel`, `parallelAxis`, `series-parallel`

- `parallel`  
Diese Komponente ist das Koordinatensystem. Eine oder mehrere Serien (wie "Beijing", "Shanghai" und "Guangzhou" im obigen Beispiel) können sich ein Koordinatensystem teilen. Wie andere Koordinatensysteme können mehrere parallele Koordinatensysteme in einer ECharts-Instanz generiert werden. Das Setzen der Position wird auch hier durchgeführt.
- `parallelAxis`  
Dies ist die Achsenkonfiguration. Mehrere Achsen sind für parallele Koordinatensysteme notwendig. `parallelAxis.parallelIndex` wird verwendet um zu spezifizieren, zu welchem Koordinatensystem die Achse gehört. Standardmäßig wird das erste Koordinatensystem verwendet.
- `series-parallel`  
Dies ist die Definition der parallelen Serien, welche im parallelen Koordinatensystem gezeichnet werden.

### Anmerkungen und Best Practices

Wenn mehrere `parallelAxis` konfiguriert werden, gibt es möglicherweise Attribute, die in jeder Achsenkonfiguration gleich gesetzt werden. Um solche Attribute nicht mehrmals schreiben zu müssen, können sie unter `parallel.parallelAxisDefault` definiert werden. Bevor die Achsen initialisiert werden, werden die Einstellungen von `parallel.parallelAxisDefault` in `parallelAxis` eingefügt, um die finale Achsenkonfiguration zu erhalten.

### Falls die Daten zu groß sind und für schlechte Performanz sorgen

Es wird empfohlen, `series-parallel.lineStyle.width` auf 0.5 (oder weniger) zu setzen, was die Performanz signifikant erhöhen kann.

### Darstellung hochdimensionaler Daten

Wenn die Dimensionszahl sehr hoch ist, sagen wir, mehr als 50 Dimensionen, dann hat das System auch mehr als 50 Achsen, was auf einer Seite kaum angezeigt werden kann.

In diesem Fall kann `parallel.axisExpandable` verwendet werden, um die Darstellung zu verbessern.

## 6.15.2 `chart.parallelAxis.id`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 6.15.3 chart.parallelAxis.dim

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Dimensionsindex der Koordinatenachse.

Zum Beispiel, sei series-parallel.data die folgenden Daten:

```
[
  [1, 55, 9, 56, 0.46, 18, 6, 'gut'],
  [2, 25, 11, 21, 0.65, 34, 9, 'exzellent'],
  [3, 56, 7, 63, 0.3, 14, 5, 'gut'],
  [4, 33, 7, 29, 0.33, 16, 6, 'exzellent'],
  { // Dateneinträge können auch Objekte sein, damit bestimmte Einstellungen der
    Zeile hier gesetzt werden können.
    value: [5, 42, 24, 44, 0.76, 40, 16, 'exzellent']
   LineStyle: {...},
  }
  ...
]
```

In den obigen Daten ist jede Zeile ein "Datenelement" und jede Spalte repräsentiert eine "Dimension". Die Bedeutungen der obigen Spalten sind zum Beispiel: "Daten", "AQI", "PM2.5", "PM10", "Level an Kohlenstoffdioxid", "Level an Stickstoffdioxid", "Level an Schwefeldioxid" und "Bewertung".

dim definiert, welche Dimension (welche Reihe) von Daten auf diese Achse gemappt wird

Start bei 0. Zum Beispiel: Falls die dim der Koordinatenachse auf 1 steht, wird die zweite Datenreihe auf diese Achse gemappt.

## 6.15.4 chart.parallelAxis.parallelIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Wird verwendet um zu definieren, in welches Koordinatensystem die Achse gemappt wird.

Zum Beispiel:

```
myChart.setOption({
  parallel: [
    {...}, // Das erste parallele Koordinatensystem
    {...} // Das zweite parallele Koordinatensystem
  ],
  parallelAxis: [
    {parallelIndex: 1, ...}, // Die erste Koordinatenachse, auf das zweite
    parallele Koordinatensystem gemappt
    {parallelIndex: 0, ...}, // Die zweite Koordinatenachse, auf das erste
    parallele Koordinatensystem gemappt
    {parallelIndex: 1, ...}, // Die dritte Koordinatenachse, auf das zweite
    parallele Koordinatensystem gemappt
    {parallelIndex: 0, ...} // Die vierte Koordinatenachse, auf das erste
    parallele Koordinatensystem gemappt
  ],
  ...
});
```

Gibt es nur ein paralleles Koordinatensystem, muss dieser Wert nicht konfiguriert werden.

### 6.15.5 chart.parallelAxis.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht aktualisiert werden soll, während die brush-Auswahl auf einer Achse verwendet wird. Steht der Wert auf false, so wird die Ansicht aktualisiert nachdem die brush-Auswahl durchgeführt wurde.

Bei großen Datenmengen wird empfohlen, diesen Wert auf false zu setzen, um Effizienzprobleme zu vermeiden.

### 6.15.6 chart.parallelAxis.areaSelectStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Bereichsauswahl auf Achsen ist möglich. Hier sind einige Konfigurationsmöglichkeiten.

### 6.15.7 chart.parallelAxis.areaSelectStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Breite der Auswahlbox.

### 6.15.8 chart.parallelAxis.areaSelectStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite der Auswahlbox

### 6.15.9 chart.parallelAxis.areaSelectStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(160,197,232)' ]

Rahmenfarbe der Auswahlbox.

### 6.15.10 chart.parallelAxis.areaSelectStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(160,197,232)' ]

Füllfarbe der Auswahlbox

### 6.15.11 chart.parallelAxis.areaSelectStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.3 ]

Undurchsichtigkeit der Auswahlbox.

### 6.15.12 chart.parallelAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'category' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheiden sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 6.15.13 chart.parallelAxis.name

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

### 6.15.14 chart.parallelAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 6.15.15 chart.parallelAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

Textstil des Achsennamens.

### 6.15.16 chart.parallelAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 6.15.17 chart.parallelAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

Erlaubte Werte

- normal
- italic
- oblique

### 6.15.18 chart.parallelAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.15.19 chart.parallelAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.15.20 chart.parallelAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 6.15.21 chart.parallelAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.15.22 chart.parallelAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.15.23 chart.parallelAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.15.24 chart.parallelAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.15.25 chart.parallelAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.15.26 chart.parallelAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.15.27 chart.parallelAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.15.28 chart.parallelAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.15.29 chart.parallelAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.15.30 chart.parallelAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.15.31 chart.parallelAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.15.32 chart.parallelAxis.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.15.33 chart.parallelAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.15.34 chart.parallelAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.15.35 chart.parallelAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.15.36 chart.parallelAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.15.37 chart.parallelAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.15.38 chart.parallelAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.15.39 chart.parallelAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.15.40 chart.parallelAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.15.41 chart.parallelAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.15.42 chart.parallelAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 6.15.43 chart.parallelAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens

### 6.15.44 chart.parallelAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 6.15.45 chart.parallelAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 6.15.46 chart.parallelAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.15.47 chart.parallelAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.15.48 chart.parallelAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 6.15.49 chart.parallelAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.15.50 chart.parallelAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.15.51 chart.parallelAxis.interval

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da `splitNumber` nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte `interval` zusammen mit `min` und `max` verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.15.52 `chart.parallelAxis.logBase`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.15.53 `chart.parallelAxis.silent`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 6.15.54 chart.parallelAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 6.15.55 chart.parallelAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Einstellungen, die zu axisLine gehören.

### 6.15.56 chart.parallelAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.15.57 chart.parallelAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.15.58 chart.parallelAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.15.59 chart.parallelAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.15.60 chart.parallelAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenlinie.

### 6.15.61 chart.parallelAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.15.62 chart.parallelAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 6.15.63 chart.parallelAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 6.15.64 chart.parallelAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.15.65 chart.parallelAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.15.66 chart.parallelAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.15.67 chart.parallelAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.15.68 chart.parallelAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.15.69 chart.parallelAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.15.70 chart.parallelAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.15.71 chart.parallelAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.15.72 chart.parallelAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.15.73 chart.parallelAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.15.74 chart.parallelAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.15.75 chart.parallelAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.15.76 chart.parallelAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, wird standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.15.77 chart.parallelAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.15.78 chart.parallelAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.15.79 chart.parallelAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.15.80 chart.parallelAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.15.81 chart.parallelAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.15.82 chart.parallelAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.15.83 chart.parallelAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.15.84 chart.parallelAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 6.15.85 chart.parallelAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.15.86 chart.parallelAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 6.15.87 chart.parallelAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.15.88 chart.parallelAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 6.15.89 chart.parallelAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 6.15.90 chart.parallelAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 6.15.91 chart.parallelAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.15.92 chart.parallelAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.15.93 chart.parallelAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.15.94 chart.parallelAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.15.95 chart.parallelAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.15.96 chart.parallelAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.15.97 chart.parallelAxis.axisLabel.fontSize

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.15.98 chart.parallelAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.15.99 chart.parallelAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.15.100 chart.parallelAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.15.101 chart.parallelAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.15.102 chart.parallelAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.15.103 chart.parallelAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.15.104 chart.parallelAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.15.105 chart.parallelAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.15.106 chart.parallelAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.15.107 chart.parallelAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.15.108 chart.parallelAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.15.109 chart.parallelAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.15.110 chart.parallelAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.15.111 chart.parallelAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.15.112 chart.parallelAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.15.113 chart.parallelAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.15.114 chart.parallelAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.15.115 chart.parallelAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.15.116 chart.parallelAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.15.117 chart.parallelAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Parallel Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.15.118 chart.parallelAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Parallel Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

## 6.16 ECharts (5) Polar

### 6.16.1 chart.polar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Polarkoordinatensysteme können in Streu- und Liniendiagrammen verwendet werden. Jedes Polarkoordinatensystem hat eine angleAxis und eine radiusAxis.

### 6.16.2 chart.polar.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.16.3 chart.polar.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plazierte, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist

zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plaziert.

#### 6.16.4 chart.polar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

z hat eine niedrigere Priorität als zlevel und erstellt keine neuen Canvas-Elemente.

#### 6.16.5 chart.polar.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Position des Zentrums vom Polarkoordinatensystem, wobei der erste Wert die horizontale Position, und der zweite die vertikale Position ist.

Prozentwerte werden unterstützt. Wenn ein Prozentwert angegeben wird, ist er beim ersten Wert relativ zur Breite des Containers, und beim zweiten relativ zur Höhe des Containers.

#### **Beispiel:**



```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 6.16.6 chart.polar.radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Radius des Polarkoordinatensystems. Als Wert sind möglich:

- Integer: Spezifiziert den äußeren Radius direkt.
- String: Zum Beispiel, '20%' bedeutet, dass der äußere Radius 20% der Größe des Viewports ist (der kleine zwischen Breite und Höhe des Chart-Containers).
- Array<Integer|String>: Der erste Wert spezifiziert den inneren Radius und der zweite Wert spezifiziert den äußeren Radius. Jeder Wert folgt den oberen Definitionen.

### 6.16.7 chart.polar.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Polarkoordinatensystems

#### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

### 6.16.8 chart.polar.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

### 6.16.9 chart.polar.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentyphen in grid und polar unterstützt. Weiterhin lassen sich Achsen über axisPointer.axis zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item

- axis
- none

### 6.16.10 chart.polar.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

tooltip.axisPointer ist wie syntaktischer Zucker von axisPointer-Einstellungen an Achsen (zum Beispiel xAxis.axisPointer oder angleAxis.axisPointer). Detailliertere Einstellungen können an someAxis.axisPointer vorgenommen werden. Im Normalfall ist die Verwendung von tooltip.axisPointer jedoch praktischer.

**Hinweis:** Konfigurationen an tooltip.axisPointer haben eine niedrigere Priorität als someAxis.axisPointer

### 6.16.11 chart.polar.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

## Erlaubte Werte

- line
- shadow
- none
- cross

### 6.16.12 chart.polar.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

## Erlaubte Werte

- x
- y
- radius
- angle
- auto

### 6.16.13 chart.polar.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

#### 6.16.14 chart.polar.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

#### 6.16.15 chart.polar.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des axisPointers.

#### 6.16.16 chart.polar.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.16.17 chart.polar.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 6.16.18 chart.polar.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.16.19 chart.polar.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 6.16.20 chart.polar.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.16.21 chart.polar.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'normal' ]

Schriftstil.

### 6.16.22 chart.polar.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.16.23 chart.polar.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.16.24 chart.polar.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße

### 6.16.25 chart.polar.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.16.26 chart.polar.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.16.27 chart.polar.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.16.28 chart.polar.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.16.29 chart.polar.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.16.30 chart.polar.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.16.31 chart.polar.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.16.32 chart.polar.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.16.33 chart.polar.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.16.34 chart.polar.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.16.35 chart.polar.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.16.36 chart.polar.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.16.37 chart.polar.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.16.38 chart.polar.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.16.39 chart.polar.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.16.40 chart.polar.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.16.41 chart.polar.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.16.42 chart.polar.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'line' ist.

### 6.16.43 chart.polar.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.16.44 chart.polar.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.16.45 chart.polar.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.16.46 chart.polar.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.16.47 chart.polar.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 6.16.48 chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.16.49 chart.polar.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.16.50 chart.polar.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.16.51 chart.polar.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

## 6.16.52 chart.polar.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.16.53 chart.polar.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.16.54 chart.polar.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 6.16.55 chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.



### 6.16.56 chart.polar.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.16.57 chart.polar.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.16.58 chart.polar.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

## 6.16.59 chart.polar.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.16.60 chart.polar.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.16.61 chart.polar.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.16.62 chart.polar.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.16.63 chart.polar.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.16.64 chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.16.65 chart.polar.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.16.66 chart.polar.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.16.67 chart.polar.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 6.16.68 chart.polar.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:



```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 6.16.69 chart.polar.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 6.16.70 chart.polar.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 6.16.71 chart.polar.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 6.16.72 chart.polar.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.16.73 chart.polar.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

### 6.16.74 chart.polar.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.16.75 chart.polar.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.16.76 chart.polar.tooltip.textStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

## 6.16.77 chart.polar.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.16.78 chart.polar.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.16.79 chart.polar.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.16.80 chart.polar.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.16.81 chart.polar.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.16.82 chart.polar.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.16.83 chart.polar.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.16.84 chart.polar.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.16.85 chart.polar.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.16.86 chart.polar.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.16.87 chart.polar.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.16.88 chart.polar.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 6.16.89 chart.polar.tooltip.axisPointer.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.16.90 chart.polar.tooltip.axisPointer.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate
- break
- breakAll

### 6.16.91 chart.polar.tooltip.textStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate

- break
- breakAll

### 6.16.92 chart.polar.tooltip.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.16.93 chart.polar.tooltip.axisPointer.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.16.94 chart.polar.tooltip.axisPointer.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.16.95 chart.polar.tooltip.axisPointer.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.16.96 chart.polar.tooltip.axisPointer.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

## Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

### 6.16.97 chart.polar.tooltip.axisPointer.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.16.98 chart.polar.tooltip.axisPointer.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 200 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.16.99 chart.polar.tooltip.axisPointer.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

### 6.16.100 chart.polar.tooltip.axisPointer.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Polar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

## 6.17 ECharts (5) Radar

### 6.17.1 chart.radar.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Koordinatensystem für Radardiagramme. Diese Komponente gleicht der Polar-Komponente in ECharts 2. Da die Polar-Komponente in ECharts 3 zur Komponente für standardmäßige Polarkoordinatensysteme umstrukturiert wurde, wurde diese Komponente zu radar umbenannt, um Verwirrung zu vermeiden.

Radarkoordinatensysteme unterscheiden sich von Polarkoordinatensystemen dadurch, dass jeder Achsenindikator des Radarkoordinatensystems eine eigene Dimension ist. Der Stil der Indikator-Koordinatenachse kann durch die folgenden Einstellungsoptionen konfiguriert werden, zusammen mit name, axisLine, axisTick, axisLabel, splitLine, splitArea.

### 6.17.2 chart.radar.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.17.3 chart.radar.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.17.4 chart.radar.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.17.5 chart.radar.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Position des Zentrums vom Polarkoordinatensystem, wobei der erste Wert die horizontale Position, und der zweite die vertikale Position ist.

Prozentwerte werden unterstützt. Wenn ein Prozentwert angegeben wird, ist er beim ersten Wert relativ zur Breite des Containers, und beim zweiten relativ zur Höhe des Containers.

#### Beispiel:

```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 6.17.6 chart.radar.radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '75%' ]

Radius des Radarkoordinatensystems. Als Wert sind möglich:

- Integer: Spezifiziert den äußeren Radius direkt.

- String: Zum Beispiel, '20%' bedeutet, dass der äußere Radius 20% der Größe des Viewports ist (der kleine zwischen Breite und Höhe des Chart-Containers).
- Array<Integer|String>: Der erste Wert spezifiziert den inneren Radius und der zweite Wert spezifiziert den äußeren Radius. Jeder Wert folgt den oberen Definitionen.

### 6.17.7 chart.radar.startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 90 ]

Der Startwinkel des Koordinatensystems, welcher der Winkel der ersten Indikatorachse ist.

### 6.17.8 chart.radar.name.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Radardiagramms.

### 6.17.9 chart.radar.name.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Name des Indikators angezeigt werden soll.

### 6.17.10 chart.radar.name.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Formatierer des Indikatornamens, welcher String-Templates und Callback-Funktionen unterstützt. Siehe das folgende Beispiel

```
// Verwendung von String-Templates, die Template-Variable {value} sollte der Name des
// Indikators sein
formatter: '{value}?'
// Verwendung von Callback-Funktionen, der erste Parameter ist der Name des
// Indikators und der zweite Parameter ist das Konfigurationsobjekt des Indikators
formatter: function (value, indicator) {
    return '?' + value + '?';
}
```

### 6.17.11 chart.radar.name.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 6.17.12 chart.radar.name.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.17.13 chart.radar.name.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.17.14 chart.radar.name.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.17.15 chart.radar.name.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.17.16 chart.radar.name.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.17.17 chart.radar.name.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.17.18 chart.radar.name.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.17.19 chart.radar.name.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.17.20 chart.radar.name.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[Default-Wert: 0]

Rahmenradius des Textfragments.

### 6.17.21 chart.radar.name.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.17.22 chart.radar.name.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.17.23 chart.radar.name.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.17.24 chart.radar.name.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.17.25 chart.radar.name.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.17.26 chart.radar.name.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.17.27 chart.radar.name.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.17.28 chart.radar.name.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.17.29 chart.radar.name.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.17.30 chart.radar.name.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.17.31 chart.radar.name.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.17.32 chart.radar.name.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.17.33 chart.radar.name.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.17.34 chart.radar.name.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.17.35 chart.radar.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Distanz zwischen Indikatorname und Achse.

### 6.17.36 chart.radar.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Segmente der Indikatorachse.

### 6.17.37 chart.radar.shape

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'polygon' ]

Rendertyp des Radars, welcher 'polygon' und 'circle' unterstützt.

Erlaubte Werte

- polygon
- circle

### 6.17.38 chart.radar.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Berechnung der Skalierung relativ zu Null verhindert werden soll. Ist der Wert auf true gesetzt, so müssen die Ticks im Koordinatensystem nicht zwingend die Null enthalten, was für gewöhnlich in Streudiagrammen mit zweifachen numerischen Wertachsen nützlich ist.

### 6.17.39 chart.radar.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

### 6.17.40 chart.radar.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```

{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}

```

### 6.17.41 chart.radar.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 6.17.42 chart.radar.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.17.43 chart.radar.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.17.44 chart.radar.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.17.45 chart.radar.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

#### 6.17.46 chart.radar.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

#### 6.17.47 chart.radar.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.17.48 chart.radar.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 1 ]

Linienbreite.

## 6.17.49 chart.radar.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

## 6.17.50 chart.radar.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.17.51 chart.radar.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.17.52 chart.radar.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.17.53 chart.radar.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.17.54 chart.radar.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.17.55 chart.radar.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.17.56 chart.radar.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.17.57 chart.radar.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinie.

### 6.17.58 chart.radar.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc'] ]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbararray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```

splitLine: {
 LineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}

```

### 6.17.59 chart.radar.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.17.60 chart.radar.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.17.61 chart.radar.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.17.62 chart.radar.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.17.63 chart.radar.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.17.64 chart.radar.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.17.65 chart.radar.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.17.66 chart.radar.splitArea.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.17.67 chart.radar.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.17.68 chart.radar.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereich der Achse im Gitterbereich, wird standardmäßig nicht angezeigt.

### 6.17.69 chart.radar.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)','rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.17.70 chart.radar.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.17.71 chart.radar.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.17.72 chart.radar.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.17.73 chart.radar.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.17.74 chart.radar.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.17.75 chart.radar.indicator.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Indikator des Radardiagramms, welcher benutzt wird um mehrere Variablen (Dimensionen) im Radardiagramm zuzuweisen. Hier ein Beispiel:

```
indicator: [
  { name: 'Sales (sales) ', max: 6500},
  { name: 'Administration (Administration) ', max: 16000, color: 'red'}, // Setze
  den Indikator auf Rot
  { name: 'Information Technology (Information Technology) ', max: 30000},
  { name: 'Customer Support (Customer Support) ', max: 38000},
  { name: 'Development (Development) ', max: 52000},
  { name: 'Marketing (Marketing) ', max: 25000}
]
```

### 6.17.76 chart.radar.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert eine Farbe für den Indikator.

### 6.17.77 chart.radar.axisName.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Namensoptionen für Radaranzeigen.

### 6.17.78 chart.radar.axisName.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Name des Indikators angezeigt werden soll.

### 6.17.79 chart.radar.axisName.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Der Formatierer des Indikatornamens, welcher String-Templates und Callback-Funktionen unterstützt. Siehe das folgende Beispiel

```
// Verwendung von String-Templates, die Template-Variable {value} sollte der Name des
// Indikators sein
formatter: '{value}?'
// Verwendung von Callback-Funktionen, der erste Parameter ist der Name des
// Indikators und der zweite Parameter ist das Konfigurationsobjekt des Indikators
formatter: function (value, indicator) {
    return '?' + value + '?';
}
```

### 6.17.80 chart.radar.axisName.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 6.17.81 chart.radar.axisName.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.17.82 chart.radar.axisName.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.17.83 chart.radar.axisName.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.17.84 chart.radar.axisName.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.17.85 chart.radar.axisName.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.17.86 chart.radar.axisName.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.17.87 chart.radar.axisName.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.17.88 chart.radar.axisName.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.17.89 chart.radar.axisName.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.17.90 chart.radar.axisName.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.17.91 chart.radar.axisName.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[Default-Wert: 0]

Rahmenradius des Textfragments.

### 6.17.92 chart.radar.axisName.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.17.93 chart.radar.axisName.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.17.94 chart.radar.axisName.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.17.95 chart.radar.axisName.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.17.96 chart.radar.axisName.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.17.97 chart.radar.axisName.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes

Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.17.98 chart.radar.axisName.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.17.99 chart.radar.axisName.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.17.100 chart.radar.axisName.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.17.101 chart.radar.axisName.textBorderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

Erlaubte Werte

- dashed
- dotted
- solid

### 6.17.102 chart.radar.axisName.textBorderDashOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderStyle` können wir den Linienstil flexibler gestalten.

### 6.17.103 `chart.radar.axisName.textShadowColor`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.17.104 `chart.radar.axisName.textShadowBlur`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 6.17.105 chart.radar.axisName.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 6.17.106 chart.radar.axisName.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.17.107 chart.radar.axisName.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

#### Erlaubte Werte

- truncate
- break
- breakAll

### 6.17.108 chart.radar.axisName.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.17.109 chart.radar.axisName.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.2.0
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.17.110 chart.radar.axisLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `borderType` können wir den Liniensstil flexibler gestalten.

### 6.17.111 chart.radar.axisLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.17.112 chart.radar.axisLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.



- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

### Erlaubte Werte

- bevel
- round
- miter

### 6.17.113 chart.radar.axisLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.17.114 chart.radar.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Einstellungen in Bezug auf Achsenticks.

### 6.17.115 chart.radar.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Setzt dies auf „false“, um zu verhindern, dass der Achsentick angezeigt wird.

### 6.17.116 chart.radar.axisTick.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenstrichs.

### 6.17.117 chart.radar.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Linienstil der Achsenstriche.

### 6.17.118 chart.radar.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die Farbe der Achsenbeschriftung ist standardmäßig auf axisLine.lineStyle.color eingestellt.

### 6.17.119 chart.radar.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

axisTick-Linienbreite.

### 6.17.120 chart.radar.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Zeilentyp. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit dashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel: { Typ: [5, 10], StrichOffset: 5 }

Erlaubte Werte

- solid
- dashed
- dotted

### 6.17.121 chart.radar.axisTick.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.17.122 chart.radar.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.17.123 chart.radar.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.17.124 chart.radar.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.17.125 chart.radar.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.17.126 chart.radar.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.17.127 chart.radar.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Einstellungen in Bezug auf die Achsenbeschriftung.

### 6.17.128 chart.radar.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 6.17.129 chart.radar.axisLabel.rotate

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Rotationsgrad der Achsenbeschriftung, was besonders nützlich ist, wenn nicht genügend Platz für die Kategorieachse vorhanden ist. Der Rotationsgrad liegt zwischen -90 und 90.

### 6.17.130 chart.radar.axisLabel.margin

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der Abstand zwischen der Achsenbeschriftung und der Achsenlinie.

### 6.17.131 chart.radar.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer der Beschriftung der Skala, unterstützt String-Templates und Callback-Funktionen. Beispiel:

```
// Verwendung von String-Templates. Die Template-Variable {value} steht für den
standardmäßigen Text der Beschriftung.
formatter: '{value} kg'

// Verwendung von Callback-Funktionen. Der Funktionsparameter ist der Wert der Skala.
formatter: function (value) {
    return value + 'km/h';
}
```

### 6.17.132 chart.radar.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.17.133 chart.radar.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.17.134 chart.radar.axisLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe

### 6.17.135 chart.radar.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 6.17.136 chart.radar.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.17.137 chart.radar.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.17.138 chart.radar.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.17.139 chart.radar.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

## Erlaubte Werte

- left
- center
- right

### 6.17.140 chart.radar.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 6.17.141 chart.radar.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.17.142 chart.radar.axisLabel.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.17.143 chart.radar.axisLabel.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.17.144 chart.radar.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.17.145 chart.radar.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.17.146 chart.radar.axisLabel.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.17.147 chart.radar.axisLabel.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.17.148 chart.radar.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.17.149 chart.radar.axisLabel.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.17.150 chart.radar.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Schattenverwischung des Textblocks.

### 6.17.151 chart.radar.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.17.152 chart.radar.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.17.153 chart.radar.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0

<b>Schlagworte / Tags</b>	
---------------------------	--

Breite des Textblocks.

### 6.17.154 chart.radar.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.17.155 chart.radar.axisLabel.textBorderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.17.156 chart.radar.axisLabel.textBorderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

Erlaubte Werte

- dashed
- dotted
- solid

### 6.17.157 chart.radar.axisLabel.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderType` können wir den Linienstil flexibler gestalten.

### 6.17.158 chart.radar.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.17.159 chart.radar.axisLabel.textShadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.17.160 chart.radar.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.17.161 chart.radar.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.17.162 chart.radar.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.17.163 chart.radar.axisLabel.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.17.164 chart.radar.axisLabel.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.17.165 chart.radar.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.17.166 chart.radar.splitLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Liniestrich-Offset einzustellen. Mit Type können wir den Liniestil flexibler gestalten.

### 6.17.167 chart.radar.splitLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.17.168 chart.radar.splitLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.



- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.17.169 chart.radar.splitLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.17.170 chart.radar.indicator.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Farbe des Indikators.

### 6.17.171 chart.radar.indicator.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Name des Indikators.

### 6.17.172 chart.radar.indicator.max

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der maximale Wert des Indikators. Es ist eine optionale Konfiguration, aber wir empfehlen, sie manuell einzustellen.

### 6.17.173 chart.radar.indicator.min

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radar
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Der Mindestwert des Indikators. Es handelt sich um eine optionale Konfiguration mit dem Standardwert 0.

## 6.18 ECharts (5) Radius Axis

### 6.18.1 chart.radiusAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Radialachse des Polarkoordinatensystems.

### 6.18.2 chart.radiusAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.18.3 chart.radiusAxis.polarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Radialachse im Polarkoordinatensystem. Ist standardmäßig die erste Achse.

### 6.18.4 chart.radiusAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

Erlaubte Werte

- value
- category
- time
- log

### 6.18.5 chart.radiusAxis.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenname.

### 6.18.6 chart.radiusAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Platzierung des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 6.18.7 chart.radiusAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 6.18.8 chart.radiusAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens, verwendet standardmäßig axisLine.lineStyle.color

### 6.18.9 chart.radiusAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

Erlaubte Werte

- normal
- italic
- oblique

### 6.18.10 chart.radiusAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.18.11 chart.radiusAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Axis-Name Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.18.12 chart.radiusAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 6.18.13 chart.radiusAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.18.14 chart.radiusAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.



Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.18.15 chart.radiusAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.18.16 chart.radiusAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.18.17 chart.radiusAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.18.18 chart.radiusAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.18.19 chart.radiusAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.18.20 chart.radiusAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].

- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.18.21 chart.radiusAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.18.22 chart.radiusAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.18.23 chart.radiusAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

#### 6.18.24 chart.radiusAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

#### 6.18.25 chart.radiusAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.18.26 chart.radiusAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.18.27 chart.radiusAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.18.28 chart.radiusAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.18.29 chart.radiusAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.18.30 chart.radiusAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.18.31 chart.radiusAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.18.32 chart.radiusAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.18.33 chart.radiusAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.18.34 chart.radiusAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 6.18.35 chart.radiusAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 6.18.36 chart.radiusAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert ist. Neue Option aus ECharts 3.

### 6.18.37 chart.radiusAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

### 6.18.38 chart.radiusAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.18.39 chart.radiusAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

## 6.18.40 chart.radiusAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

## 6.18.41 chart.radiusAxis.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente, in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.18.42 chart.radiusAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.18.43 chart.radiusAxis.interval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

#### 6.18.44 chart.radiusAxis.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

#### 6.18.45 chart.radiusAxis.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

#### 6.18.46 chart.radiusAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

### 6.18.47 chart.radiusAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 6.18.48 chart.radiusAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.18.49 chart.radiusAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.18.50 chart.radiusAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.



### 6.18.51 chart.radiusAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.18.52 chart.radiusAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

### 6.18.53 chart.radiusAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333 ]

## Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.18.54 chart.radiusAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.18.55 chart.radiusAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dotted
- dashed

### 6.18.56 chart.radiusAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.57 chart.radiusAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.58 chart.radiusAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.59 chart.radiusAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.60 chart.radiusAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.61 chart.radiusAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.18.62 chart.radiusAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.18.63 chart.radiusAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.18.64 chart.radiusAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.18.65 chart.radiusAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.18.66 chart.radiusAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.18.67 chart.radiusAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 6.18.68 chart.radiusAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks, ist standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.18.69 chart.radiusAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.18.70 chart.radiusAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.18.71 chart.radiusAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.72 chart.radiusAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.73 chart.radiusAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.74 chart.radiusAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.75 chart.radiusAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.76 chart.radiusAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen der Achsenbeschriftung.

### 6.18.77 chart.radiusAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.18.78 chart.radiusAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.18.79 chart.radiusAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.18.80 chart.radiusAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 6.18.81 chart.radiusAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 6.18.82 chart.radiusAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```

// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
// Achsenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
// Achsenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getFullYear());
  }
  return texts.join('/');
}

```

### 6.18.83 chart.radiusAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.18.84 chart.radiusAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.18.85 chart.radiusAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.18.86 chart.radiusAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.18.87 chart.radiusAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.18.88 chart.radiusAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.18.89 chart.radiusAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.18.90 chart.radiusAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 6.18.91 chart.radiusAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.18.92 chart.radiusAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.18.93 chart.radiusAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.18.94 chart.radiusAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.18.95 chart.radiusAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.18.96 chart.radiusAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.18.97 chart.radiusAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.18.98 chart.radiusAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.18.99 chart.radiusAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.18.100 chart.radiusAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.18.101 chart.radiusAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.18.102 chart.radiusAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.18.103 chart.radiusAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.18.104 chart.radiusAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.18.105 chart.radiusAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.18.106 chart.radiusAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.18.107 chart.radiusAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.18.108 chart.radiusAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.18.109 chart.radiusAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.18.110 chart.radiusAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.18.111 chart.radiusAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.18.112 chart.radiusAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.18.113 chart.radiusAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.18.114 chart.radiusAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 6.18.115 chart.radiusAxis.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbararray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 6.18.116 chart.radiusAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinie.

### 6.18.117 chart.radiusAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.18.118 chart.radiusAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.119 chart.radiusAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.120 chart.radiusAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.121 chart.radiusAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.122 chart.radiusAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.123 chart.radiusAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.18.124 chart.radiusAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'auto' ]

### 6.18.125 chart.radiusAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.18.126 chart.radiusAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 6.18.127 chart.radiusAxis.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.18.128 chart.radiusAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.129 chart.radiusAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.130 chart.radiusAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.131 chart.radiusAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.132 chart.radiusAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.133 chart.radiusAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 6.18.134 chart.radiusAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 6.18.135 chart.radiusAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatorotyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.18.136 chart.radiusAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.18.137 chart.radiusAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.18.138 chart.radiusAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.18.139 chart.radiusAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.18.140 chart.radiusAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

### 6.18.141 chart.radiusAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Der String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 6.18.142 chart.radiusAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Der Abstand zwischen dem Label und der Achse.

### 6.18.143 chart.radiusAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.18.144 chart.radiusAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 6.18.145 chart.radiusAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.18.146 chart.radiusAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.18.147 chart.radiusAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.18.148 chart.radiusAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.18.149 chart.radiusAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.18.150 chart.radiusAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.18.151 chart.radiusAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.18.152 chart.radiusAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.18.153 chart.radiusAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.18.154 chart.radiusAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.18.155 chart.radiusAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.18.156 chart.radiusAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.18.157 chart.radiusAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.18.158 chart.radiusAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.18.159 chart.radiusAxis.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.18.160 chart.radiusAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.18.161 chart.radiusAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.162 chart.radiusAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.163 chart.radiusAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.164 chart.radiusAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.165 chart.radiusAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 6.18.166 chart.radiusAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

## 6.18.167 chart.radiusAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

## 6.18.168 chart.radiusAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

## 6.18.169 chart.radiusAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.170 chart.radiusAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.171 chart.radiusAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.172 chart.radiusAxis.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.173 chart.radiusAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.174 chart.radiusAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.18.175 chart.radiusAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.18.176 chart.radiusAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.177 chart.radiusAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.178 chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.18.179 chart.radiusAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 6.18.180 chart.radiusAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.18.181 chart.radiusAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.18.182 chart.radiusAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.18.183 chart.radiusAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 6.18.184 chart.radiusAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.18.185 chart.radiusAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um das Handle zu verwenden.

### 6.18.186 chart.radiusAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Icon des Handles.

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.18.187 chart.radiusAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.18.188 chart.radiusAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.18.189 chart.radiusAxis.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.18.190 chart.radiusAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.18.191 chart.radiusAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.192 chart.radiusAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.193 chart.radiusAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.194 chart.radiusAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.195 chart.radiusAxis.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.18.196 chart.radiusAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.18.197 chart.radiusAxis.nameTextStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.18.198 chart.radiusAxis.nameTextStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.18.199 chart.radiusAxis.axisLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.200 chart.radiusAxis.axisLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.18.201 chart.radiusAxis.axisLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.18.202 chart.radiusAxis.axisLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity - und NaN-Werte werden ignoriert.

### 6.18.203 chart.radiusAxis.axisTick.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.204 chart.radiusAxis.axisTick.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.18.205 chart.radiusAxis.axisTick.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.18.206 chart.radiusAxis.splitLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity - und NaN-Werte werden ignoriert.

### 6.18.207 chart.radiusAxis.splitLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.208 chart.radiusAxis.splitLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.18.209 chart.radiusAxis.splitLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.18.210 chart.radiusAxis.axisPointer.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

Erlaubte Werte

- truncate
- break
- breakAll

### 6.18.211 chart.radiusAxis.axisPointer.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.18.212 chart.radiusAxis.axisPointer.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.213 chart.radiusAxis.axisPointer.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt



Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.18.214 chart.radiusAxis.axisPointer.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.18.215 chart.radiusAxis.axisPointer.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity - und NaN-Werte werden ignoriert.

### 6.18.216 chart.radiusAxis.minorTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Einstellungen im Zusammenhang mit kleinen Häkchen. Hinweis: minorTick ist in der Kategorietyppachse nicht verfügbar.

### 6.18.217 chart.radiusAxis.minorTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die MinorAxis angezeigt werden soll.

### 6.18.218 chart.radiusAxis.minorTick.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl des Intervalls geteilt durch kleine Ticks.

### 6.18.219 chart.radiusAxis.minorTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Länge der kleinen Tick-Linien.

### 6.18.220 chart.radiusAxis.minorTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des MinorTick-Linienstils

### 6.18.221 chart.radiusAxis.minorTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stilkonfiguration kleiner Teilstriche

### 6.18.222 chart.radiusAxis.minorTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der MinorTicks.

### 6.18.223 chart.radiusAxis.minorTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Zeilentyp. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit dashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ type: [5, 10], dashOffset: 5 }
```

Erlaubte Werte

- solid
- dashed
- dotted

### 6.18.224 chart.radiusAxis.minorTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.225 chart.radiusAxis.minorTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.226 chart.radiusAxis.minorTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.227 chart.radiusAxis.minorTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.18.228 chart.radiusAxis.minorTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.18.229 chart.radiusAxis.minorTick.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.230 chart.radiusAxis.minorTick.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

## Erlaubte Werte

- butt
- round
- square

### 6.18.231 chart.radiusAxis.minorTick.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.18.232 chart.radiusAxis.minorTick.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0



### Schlagworte / Tags

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

## 6.18.233 chart.radiusAxis.minorSplitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Kleinere Teilungslinien der Achse im Rasterbereich?Es wird am minorTick ausgerichtet

## 6.18.234 chart.radiusAxis.minorSplitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Minor-Split-Lines angezeigt werden soll.

## 6.18.235 chart.radiusAxis.minorSplitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des MinorSplitLine -Liniensstils

### 6.18.236 chart.radiusAxis.minorSplitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

minorSplitLineLine Farbe

### 6.18.237 chart.radiusAxis.minorSplitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der minorSplitLines.

### 6.18.238 chart.radiusAxis.minorSplitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Zeilentyp. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit dashOffset können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ type: [5, 10], dashOffset: 5 }
```

Erlaubte Werte

- solid
- dashed
- dotted

### 6.18.239 chart.radiusAxis.minorSplitLine.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.18.240 chart.radiusAxis.minorSplitLine.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.18.241 chart.radiusAxis.minorSplitLine.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

## 6.18.242 chart.radiusAxis.minorSplitLine.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

## 6.18.243 chart.radiusAxis.minorSplitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.18.244 chart.radiusAxis.minorSplitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.18.245 chart.radiusAxis.minorSplitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.18.246 chart.radiusAxis.minorSplitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 6.18.247 chart.radiusAxis.minorSplitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Radius Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.19 ECharts (5) Series

### 6.19.1 chart.series.i[].activeOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Bei der brush-Auswahl werden asugewählte Linien auf diese Transparenz gesetzt (was diese Linien hervorheben kann).

### 6.19.2 chart.series.i[].animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.19.3 chart.series.i[].animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.19.4 chart.series.i[].animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

## 6.19.5 chart.series.i[].animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

## 6.19.6 chart.series.i[].animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.7 chart.series.i[].animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn

- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.8 chart.series.i[].animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut

- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.9 chart.series.i[].animationEasingUpdate.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Zeitverzögerung bevor die Animation startet. Callback-Funktionen können verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
  return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
  return (params.count - 1 - params.index) * 30;
}
```

### 6.19.10 chart.series.i[].animationEasingUpdate.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung vor der Aktualisierung der Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben.

Zum Beispiel:

```
animationDelayUpdate: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.19.11 chart.series.i[.animationEasingUpdate.object

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut

- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.12 chart.series.i[].animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.19.13 chart.series.i[].animationType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'expansion' ]

Initialer Animationstyp.

- 'expansion' Standardmäßige expandierende Animation.
- 'scale' Skalierende Animation. Kann zusammen mit `animationEasing='elasticOut'` verwendet werden um einen Popup-Effekt zu haben.

Erlaubte Werte

- expansion
- scale

### 6.19.14 chart.series.i[[]].areaStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel `'rgb(128, 128, 128)'`. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel `'rgba(128, 128, 128, 0.5)'`. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel `'#ccc'`. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.15 chart.series.i[.].areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

Der Stil des Bereichs.

## 6.19.16 chart.series.i[].areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.19.17 chart.series.i[].areaStyle.origin

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Ursprungsposition des Bereichs.

Standardmäßig ist der Bereich zwischen der Achsenlinie und den Daten der Bereich der gefüllt wird. Diese Einstellung erlaubt es, von den Daten zum Maximal- oder Minimalwert der Achse zu füllen.

Mögliche Werte:

- 'auto': Um von der Achsenlinie bis zu den Daten zu füllen;
- 'start': Um vom Minimalwert der Achse bis zu den Daten zu füllen;
- 'end': Um vom Maximalwert der Achse bis zu den Daten zu füllen;

Erlaubte Werte

- auto
- start

- end

### 6.19.18 chart.series.i[.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.19 chart.series.i[.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.20 chart.series.i[.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.21 chart.series.i[[]].areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.22 chart.series.i[[]].aspectScale

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.75 ]

Wird verwendet, um das Seitenverhältnis von geo zu skalieren.

Das endültige Seitenverhältnis berechnet sich aus:  $\text{geoBoundingRect.width} / \text{geoBoundingRect.height} * \text{aspectScale}$ .

### 6.19.23 chart.series.i[].avoidLabelOverlap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Strategie zum Vermeiden sich überlappender Beschriftungen aktiviert werden soll. Standardmäßig aktiviert, wodurch Beschriftungen auseinanderbewegt werden, wenn sie sich überlappen.

### 6.19.24 chart.series.i[].axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.25 chart.series.i[].axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.26 chart.series.i[].axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.27 chart.series.i[].axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.28 chart.series.i[].axisLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe

### 6.19.29 chart.series.i[].axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.30 chart.series.i[].axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.31 chart.series.i[].axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.32 chart.series.i[].axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.19.33 chart.series.i[].axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer der Beschriftung der Skala, unterstützt String-Templates und Callback-Funktionen. Beispiel:

```
// Verwendung von String-Templates. Die Template-Variable {value} steht für den
// standardmäßigen Text der Beschriftung.
formatter: '{value} kg'

// Verwendung von Callback-Funktionen. Der Funktionsparameter ist der Wert der Skala.
formatter: function (value) {
  return value + 'km/h';
}
```

### 6.19.34 chart.series.i[].axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.35 chart.series.i[].axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.36 chart.series.i[].axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Beschriftung der Achsenticks.

### 6.19.37 chart.series.i[].axisLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.38 chart.series.i[].axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.39 chart.series.i[.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.40 chart.series.i[].axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.41 chart.series.i[].axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.42 chart.series.i[].axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.43 chart.series.i[].axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 6.19.44 chart.series.i[].axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.45 chart.series.i[].axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 6.19.46 chart.series.i[].axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

#### 6.19.47 chart.series.i[].axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

#### 6.19.48 chart.series.i[].axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

#### 6.19.49 chart.series.i[].axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

#### 6.19.50 chart.series.i[].axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.51 chart.series.i[].axisLine.lineStyle.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-wert: [[0.2, '#91c7ae'], [0.8, '#63869e'], [1, '#c23531']] ]

Die Achsenlinie des Messdiagramms kann in mehrere Segmente mit verschiedenen Farben aufgeteilt werden. Die Endposition und Farbe jedes Segments kann in einem Array dargestellt werden.

### 6.19.52 chart.series.i[].axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie des Messdiagramms.

### 6.19.53 chart.series.i[].axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.19.54 chart.series.i[].axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.19.55 chart.series.i[].axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.56 chart.series.i[].axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.57 chart.series.i[].axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.58 chart.series.i[].axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Linienbreite.

### 6.19.59 chart.series.i[].axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenlinie des Messdiagramms gehören.

### 6.19.60 chart.series.i[].axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie des Messdiagramms angezeigt werden soll.

### 6.19.61 chart.series.i[].axisTick.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 8 ]

Die Länge der Ticklinie, kann ein Prozentwert relativ zu radius sein.

### 6.19.62 chart.series.i[].axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#eee' ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.63 chart.series.i[[]].axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenticks.

### 6.19.64 chart.series.i[].axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.65 chart.series.i[].axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.66 chart.series.i[].axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.67 chart.series.i[].axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.68 chart.series.i[].axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.69 chart.series.i[].axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.70 chart.series.i[].axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.19.71 chart.series.i[].axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Ticklinientyp.

### 6.19.72 chart.series.i[].axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Skala angezeigt werden soll.

### 6.19.73 chart.series.i[].axisTick.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Anzahl der durch Achsenticks getrennten Segmente zwischen den Trennlinien.

### 6.19.74 chart.series.i[].barCategoryGap

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '20%' ]

Der Säulenabstand einer einzelnen Serie, standardmäßig 20% des Kategorienabstands, kann als fester Wert gesetzt werden.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie im Koordinatensystem gesetzt werden, und wird dann von allen 'bar'-Serien im Koordinatensystem übernommen.

### 6.19.75 chart.series.i[].barGap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30% ]

Der Abstand zwischen Säulen verschiedener Serien, ist ein Prozentwert wie '30%', was für 30% der Breite der Säule steht.

Setzen von barGap auf '-100%' lässt Säulen verschiedener Serien überlappen, was nützlich ist, wenn eine Säulenserie als Hintergrund dienen soll.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie des Koordinatensystems gesetzt werden, und wird dann von allen 'bar'-Serien im koordinatensystem übernommen.

### 6.19.76 chart.series.i[].barMaxWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Maximale Breite der Säulen. Absolute Werte (wie 10) und Prozentwerte (wie '20%', nach der Breite des Diagramms) können verwendet werden. Adaptiv wenn nicht gesetzt.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie im Koordinatensystem gesetzt werden, und wird dann von allen 'bar'-Serien im Koordinatensystem übernommen.

### 6.19.77 chart.series.i[].barMinHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Minimalbreite der Säule. Kann verwendet werden, um die Situation zu vermeiden, dass Interaktionen durch Säulen mit zu kleinen Datenwerten beeinträchtigt werden.

### 6.19.78 chart.series.i[].barMinWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert minimale Breite der Säulen. Absolute Werte (wie 10) oder Prozentwerte (wie '20%', nach der Breite des Diagramms) können verwendet werden. Passt sich standardmäßig automatisch an.

## Erlaubte Werte

- {integer}
- {percent}

### 6.19.79 chart.series.i[].barWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Breite der Säule. Adaptiv wenn nicht spezifiziert.

In einem einzelnen Koordinatensystem wird sich dieses Attribut von mehreren 'bar'-Serien geteilt. Dieses Attribut sollte für die letzte 'bar'-Serie des Koordinatensystems gesetzt werden, und wird dann von allen 'bar'-Serien im koordinatensystem übernommen.

### 6.19.80 chart.series.i[].blurSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Verwischungsgröße der Datenpunkte. Ist gültig wenn coordinateSystem den Wert 'geo' hat.

### 6.19.81 chart.series.i[].bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der Unterseite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein, oder ein prozentualer Wert relativ zur Containerbreite wie '20%'.

Erlaubte Werte

- {integer}
- {percent}

### 6.19.82 chart.series.i[].boundaryGap

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['10%', '10%'] ]

Der Grenzabstand der Richtung, die Orthogonal zur KoordinatenAchse im Diagramm ist. Wird gesetzt, um die Position des Diagramms anzupassen, und es im Mittelpunkt des Bildschirms statt des oberen oder unteren Bereichs zu halten.

Erlaubte Werte

- {integer}
- {percent}

### 6.19.83 chart.series.i[].boundingCoords

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zweidimensionales Array. Definiert die Koordinaten von oben-links und unten-rechts in der Layout-Box

```
// Eine komplette Weltkarte
map: 'world',
left: 0, top: 0, right: 0, bottom: 0,
boundingCoords: [
  // [Längengrad, Breitengrad] der Ecke oben-links
  [-180, 90],
  // [Längengrad, Breitengrad] der Ecke unten-rechts
  [180, -90]
],
```

#### 6.19.84 chart.series.i[].boxWidth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [7, 50] ]

Obere und Untere Grenze der Größe der Box. Das Array hat die Form [min, max].

Möglich sind absolute Pixelwerte, wie [7, 50], oder Prozentwerte, wie ['40%', '90%']. Die Prozentwerte sind hierbei relativ zur maximal möglichen Breite.

#### 6.19.85 chart.series.i[].breadcrumb.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der Unterseite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerhöhe wie '20%'.

Erlaubte Werte

- {integer}
- {percent}

### 6.19.86 chart.series.i[].breadcrumb.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(255,255,255,0.7) ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.87 chart.series.i[].breadcrumb.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

## Erlaubte Werte

- dashed
- dotted
- solid

## 6.19.88 chart.series.i[].breadcrumb.emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 6.19.89 chart.series.i[].breadcrumb.emphasis.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(0,0,0,0.7) ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.90 chart.series.i[].breadcrumb.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

## 6.19.91 chart.series.i[].breadcrumb.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.19.92 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.93 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(150,150,150,1) ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.94 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.95 chart.series.i[].breadcrumb.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.96 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.97 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.98 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.99 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.100 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.101 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Textfarbe.

### 6.19.102 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.103 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.104 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 6.19.105 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.106 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.107 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.108 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des hervorgehobenen Breadcrumb-Textes.

### 6.19.109 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.110 chart.series.i[.breadcrumb.emphasis.itemStyle.textStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.111 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

6.19.112 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowColor`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

6.19.113 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetX`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

6.19.114 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.shadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.115 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes.

### 6.19.116 chart.series.i[[]].breadcrumb.emphasis.itemStyle.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes.

## 6.19.117 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 6.19.118 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.19.119 chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.120 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.textShadowOffsetX`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.121 `chart.series.i[].breadcrumb.emphasis.itemStyle.textStyle.verticalAlign`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im rich nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

### Erlaubte Werte

- top
- middle
- bottom

## 6.19.122 chart.series.i[.].breadcrumb.emphasis.itemStyle.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### Erlaubte Werte

- {percent}



### 6.19.123 chart.series.i[].breadcrumb.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausgewähltes Element des Breadcrumbs.

### 6.19.124 chart.series.i[].breadcrumb.emptyItemWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 25 ]

Wenn der Breadcrumb keinen Inhalt hat, wird diese Minimalbreite verwendet.

### 6.19.125 chart.series.i[].breadcrumb.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 22 ]

Die Höhe des Breadcrumbs.

### 6.19.126 chart.series.i[.].breadcrumb.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(255,255,255,0.7) ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.127 chart.series.i[.].breadcrumb.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.128 chart.series.i[.].breadcrumb.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.19.129 chart.series.i[.].breadcrumb.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(0,0,0,0.7) ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.130 chart.series.i[.].breadcrumb.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

### 6.19.131 chart.series.i[.].breadcrumb.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.132 chart.series.i[.].breadcrumb.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.133 chart.series.i[.].breadcrumb.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: rgba(150,150,150,1) ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.134 chart.series.i[.].breadcrumb.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.135 chart.series.i[.].breadcrumb.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.136 chart.series.i[.].breadcrumb.itemStyle.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.137 chart.series.i[.].breadcrumb.itemStyle.textStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.138 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.139 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.140 chart.series.i[.].breadcrumb.itemStyle.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.141 chart.series.i[.].breadcrumb.itemStyle.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #fff ]

Textfarbe.

### 6.19.142 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.143 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.144 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 6.19.145 chart.series.i[.].breadcrumb.itemStyle.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.146 chart.series.i[.].breadcrumb.itemStyle.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.147 chart.series.i[.breadcrumb.itemStyle.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.148 chart.series.i[.breadcrumb.itemStyle.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Breadcrumb-Textes.

### 6.19.149 chart.series.i[.breadcrumb.itemStyle.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.150 chart.series.i[.breadcrumb.itemStyle.textStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.151 chart.series.i[.].breadcrumb.itemStyle.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.152 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.153 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.19.154 chart.series.i[].breadcrumb.itemStyle.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.155 chart.series.i[.].breadcrumb.itemStyle.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes.

### 6.19.156 chart.series.i[.].breadcrumb.itemStyle.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes.

### 6.19.157 chart.series.i[.].breadcrumb.itemStyle.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.158 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.159 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.160 chart.series.i[].breadcrumb.itemStyle.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.161 chart.series.i[.].breadcrumb.itemStyle.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.162 chart.series.i[.].breadcrumb.itemStyle.textStyle.width

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.19.163 chart.series.i[.].breadcrumb.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Abstand zwischen der Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerbreite wie '20%'. Er kann auch 'left', 'center' oder 'right' sein.

Steht der Wert von left auf 'left', 'center' oder 'right', wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}

- {percent}
- left
- center
- right

### 6.19.164 chart.series.i[.].breadcrumb.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breadcrumb, zeigt den Pfad des momentanen Knotenpunkts an.

### 6.19.165 chart.series.i[.].breadcrumb.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Komponente und der rechten Seite des Containers.

Der Wert von right kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerbreite wie '20%'.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {integer}
- {percent}

### 6.19.166 chart.series.i[[]].breadcrumb.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Breadcrumb angezeigt werden soll.

### 6.19.167 chart.series.i[[]].breadcrumb.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein, oder ein Prozentwert relativ zur Containerhöhe wie '20%'. Er kann auch 'top', 'middle' oder 'bottom' sein.

Steht der Wert von top auf 'top', 'middle' oder 'bottom', wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.19.168 chart.series.i[].calendarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Kalenderkoordinaten mit denen kombiniert werden soll, nützlich für mehrere Kalenderkoordinaten in einem Diagramm.

### 6.19.169 chart.series.i[].categories.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.170 chart.series.i[].categories.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.171 chart.series.i[.].categories.emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.172 chart.series.i[.].categories.emphasis.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.173 chart.series.i[.].categories.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



**Schlagworte / Tags**

Grafischer Stil der Elemente, emphasis ist der Stil im hervorgehobenen Zustand, zum Beispiel durch die Maus oder über die Legendenverbindung.

### 6.19.174 chart.series.i[].categories.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.175 chart.series.i[].categories.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.176 chart.series.i[].categories.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.177 chart.series.i[].categories.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.178 chart.series.i[].categories.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.179 chart.series.i[].categories.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.180 chart.series.i[].categories.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.181 chart.series.i[].categories.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.182 chart.series.i[].categories.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.183 chart.series.i[].categories.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.184 chart.series.i[].categories.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.185 chart.series.i[].categories.emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.186 chart.series.i[].categories.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.187 chart.series.i[].categories.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.188 chart.series.i[].categories.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.189 chart.series.i[].categories.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.190 chart.series.i[].categories.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:



```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.191 chart.series.i[.].categories.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.192 chart.series.i[.].categories.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.193 chart.series.i[].categories.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Ausgewählten Knotens.

### 6.19.194 chart.series.i[].categories.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 6.19.195 chart.series.i[].categories.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.196 chart.series.i[].categories.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.197 chart.series.i[.categories.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.198 chart.series.i[.].categories.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.199 chart.series.i[].categories.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.200 chart.series.i[].categories.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.201 chart.series.i[].categories.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.202 chart.series.i[].categories.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.203 chart.series.i[].categories.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.204 chart.series.i[].categories.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.205 chart.series.i[].categories.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.206 chart.series.i[].categories.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.207 chart.series.i[].categories.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.208 chart.series.i[].categories.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.209 chart.series.i[].categories.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.210 chart.series.i[].categories.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.211 chart.series.i[].categories.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.212 chart.series.i[].categories.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausgewählte Kategorie.

### 6.19.213 chart.series.i[].categories.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.214 chart.series.i[].categories.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.215 chart.series.i[.].categories.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.216 chart.series.i[.].categories.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.217 chart.series.i[].categories.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Knoten dieser Kategorie.

### 6.19.218 chart.series.i[].categories.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.219 chart.series.i[].categories.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.220 chart.series.i[].categories.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.221 chart.series.i[].categories.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.222 chart.series.i[].categories.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.223 chart.series.i[.].categories.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.224 chart.series.i[.].categories.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.225 chart.series.i[].categories.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.226 chart.series.i[].categories.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.227 chart.series.i[].categories.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.228 chart.series.i[].categories.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.229 chart.series.i[].categories.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.230 chart.series.i[].categories.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.231 chart.series.i[].categories.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.232 chart.series.i[.categories.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.233 chart.series.i[.categories.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.234 chart.series.i[].categories.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.235 chart.series.i[].categories.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.236 chart.series.i[].categories.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.237 chart.series.i[].categories.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Beschriftungen der Knoten dieser Kategorie.

### 6.19.238 chart.series.i[].categories.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 6.19.239 chart.series.i[].categories.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.240 chart.series.i[].categories.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:



```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.241 chart.series.i[.].categories.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.242 chart.series.i[.].categories.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.243 chart.series.i[].categories.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.244 chart.series.i[].categories.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.245 chart.series.i[].categories.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.246 chart.series.i[].categories.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.247 chart.series.i[].categories.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.248 chart.series.i[].categories.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.249 chart.series.i[].categories.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.250 chart.series.i[].categories.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.251 chart.series.i[].categories.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.252 chart.series.i[.].categories.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.253 chart.series.i[.].categories.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.254 chart.series.i[].categories.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.255 chart.series.i[].categories.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.256 chart.series.i[].categories.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Kategorie, welcher mit der Legende und den Beschriftungen korrespondieren kann.

### 6.19.257 chart.series.i[].categories.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Kategorisierung der Knoten, welche optional ist. Müssen Knoten klassifiziert werden, so lässt sich die Kategorie jedes Knotens über data[i].category festlegen. Der Stil der Kategorie wird auch auf de Knoten der Kategorie angewendet. Kategorien können auch in der Legende verwendet werden.

### 6.19.258 chart.series.i[].categories.symbol

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Knoten dieser Kategorie.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

#### 6.19.259 chart.series.i[].categories.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 6.19.260 chart.series.i[].categories.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 6.19.261 chart.series.i[].categories.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 6.19.262 chart.series.i[].categories.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße der Knoten dieser Kategorie. Kann auf eine einzelne Zahl wie 10 gesetzt werden, oder auf ein Array um Breite und Höhe darzustellen. Zum Beispiel, [20, 10] steht für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

### 6.19.263 chart.series.i[.center

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['50%', '50%'] ]

Zentrale Position des Diagramms, der erste Wert steht für die horizontale, der zweite für die vertikale Position.

Prozentwerte werden unterstützt. Werden Prozentwerte angegeben, so sind diese relativ zu jeweils der Containerbreite und Containerhöhe.

#### Beispiel:

```
// Auf absolute Pixelwerte setzen
center: [400, 300]
// Auf relative Prozentwerte setzen
center: ['50%', '50%']
```

### 6.19.264 chart.series.i[.childrenVisibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Kinder werden nicht angezeigt wenn die Größe des Bereichs eines Knotens kleiner als dieser Wert ist (Einheit: Quadratpixel).

Durch diese Einstellung können die Details von Knoten versteckt werden, wenn der rechteckige Bereich nicht groß genug ist. Wenn Nutzer auf den Knoten hereinzoomen, werden die Kindknoten angezeigt wenn der Bereich größer als der angegebene Wert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann das `childrenVisibleMin` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.265 `chart.series.i[].circular.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfiguration des zirkulären Layouts.

### 6.19.266 `chart.series.i[].circular.rotateLabel`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen automatisch gedreht werden sollen.

### 6.19.267 chart.series.i[].clipOverflow

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der überstehende Teil abgeschnitten werden soll, standardmäßig wird abgeschnitten.

### 6.19.268 chart.series.i[].clockwise

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob im Uhrzeigersinn gezeichnet werden soll.

### 6.19.269 chart.series.i[].colorAlpha

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Transparenzraten (Alpha-Kanal) der Serie an. Akzeptiert Werte von 0 bis 1.

Der Wert kann zum Beispiel [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorAlpha` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.270 `chart.series.i[].colorMappingBy`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'index' ]

Spezifiziert die Regel, nach der jeder Knoten seine Farbe aus der Farbliste erhält. Mögliche Werte:

- 'value':

Mappt `series-treemap.data.value` auf die Farben.

So zeigen die Farben der Knoten ihre Werte an.

`visualDimension` kann verwendet werden, um zu bestimmen, welche Datendimension für das visuelle Mapping verwendet wird.

- 'index':

Mappt den Index (Ordinalzahl) des Knotens auf die Farben. Dadurch bekommt in einer Ebene der erste Knoten die erste Farbe der Liste und der zweite Knoten die zweite Farbe.

So können aufeinanderfolgende Knoten durch ihre Farben unterschieden werden.

- 'id':

Mappt `series-treemap.data.id` auf die Farben.

Da die `id` verwendet wird um die Knoten zu identifizieren, behalten die Knoten so ihre ursprüngliche Farbe, wenn der Baum durch `setOption` verändert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

## Erlaubte Werte

- `value`
- `index`
- `id`

### 6.19.271 `chart.series.i[].colorSaturation`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Sättigungswerten der Serie an. Akzeptiert Werte von 0 bis 1.

Zum Beispiel kann `colorSaturation` [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.272 `chart.series.i[].connectNulls`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linie über Nullpunkte verbunden werden soll.

### 6.19.273 chart.series.i[].coordinateSystem

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cartesian2d' ]

Das Koordinatensystem, das für die Serie verwendet wird. Mögliche Werte sind:

- 'cartesian2d'  
Verwendet ein zweidimensionales, rechteckiges Koordinatensystem (auch kartesisches Koordinatensystem genannt), wobei die zugehörigen Achsenkomponenten über xAxisIndex und yAxisIndex zugewiesen werden.
- 'polar'  
Verwendet ein Polarkoordinatensystem, wobei die zugehörige Polarkoordinaten-Komponente über polarIndex zugewiesen wird.

#### Erlaubte Werte

- cartesian2d
- polar

### 6.19.274 chart.series.i[].cursor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'pointer' ]

Der Mausstil wenn die Maus auf ein Element zeigt, gleicht der cursor-Eigenschaft in CSS.



### 6.19.275 chart.series.i[].datasetIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Ist series.data nicht spezifiziert, und dataset existiert, so verwendet die Serie dataset. dataSetIndex spezifiziert, welches dataset verwendet wird.

### 6.19.276 chart.series.i[].detail.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.277 chart.series.i[].detail.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.278 chart.series.i[].detail.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.279 chart.series.i[].detail.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.280 chart.series.i[].detail.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Textfarbe.

### 6.19.281 chart.series.i[].detail.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.282 chart.series.i[].detail.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Schriftgröße.

### 6.19.283 chart.series.i[.].detail.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.19.284 chart.series.i[.].detail.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.19.285 chart.series.i[.].detail.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.286 chart.series.i[.].detail.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.287 chart.series.i[].detail.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Details des Messdiagramms, welche zum Anzeigen von Daten verwendet werden.

### 6.19.288 chart.series.i[].detail.offsetCenter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, '40%'] ]

Die Offset-Position relativ zum Mittelpunkt des Messdiagramms. Das erste Element des Arrays ist der horizontale Offset, das zweite Element ist der vertikale Offset. Kann in absoluten Werten oder Prozentwerten relativ zum Radius des Messdiagramms angegeben werden.

## Erlaubte Werte

- {integer}
- {percent}

### 6.19.289 chart.series.i[.].detail.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.290 chart.series.i[.].detail.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.291 chart.series.i[.].detail.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.



## 6.19.292 chart.series.i[].detail.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.293 chart.series.i[].detail.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.19.294 chart.series.i[].detail.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.295 chart.series.i[].detail.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Details angezeigt werden sollen.

### 6.19.296 chart.series.i[].detail.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.297 chart.series.i[].detail.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.298 chart.series.i[.].detail.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.299 chart.series.i[.].detail.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.300 chart.series.i[.].detail.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.301 chart.series.i[.].detail.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.302 chart.series.i[.].detail.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.303 chart.series.i[].dimensions

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

dimensions kann verwendet werden um Informationen für die Dimensionen von series.data oder dataset.source zu definieren.

Hinweis: Wird dataset verwendet, so können wir Dimensionsnamen in der ersten Reihe/Spalte von dataset.source bereitstellen, und müssen dimensions nicht hier spezifizieren. Wird dimensions jedoch hier spezifiziert, so wird ECharts keine Dimensionsnamen mehr aus der ersten Reihe/Spalte von dataset.source holen.

Zum Beispiel:

```
option = {
  dataset: {
    source: [
      // 'date', 'open', 'close', 'highest', 'lowest'
      [12, 44, 55, 66, 2],
      [23, 6, 16, 23, 1],
      ...
    ]
  },
  series: {
    type: 'xxx',
    // Spezifiziere die Dimensionsnamen, welche im Tooltip angezeigt werden.
    dimensions: ['date', 'open', 'close', 'highest', 'lowest']
  }
}
```

```

series: {
  type: 'xxx',
  dimensions: [
    null, // Soll eine Dimension nicht definiert werden, ist null
    // Typ der Dimension spezifizieren.
    // 'ordinal' wird immer in Strings verwendet.
    // Ist der Typ nicht spezifiziert, so errät ECharts den
    // Typ anhand der Daten
    {name: 'good', type: 'number'},
    'bad' // Ist gleich zu {name: 'bad'}.
  ]
}

```

Jeder Eintrag von dimensions kann sein:

- String, zum Beispiel 'someName', was gleich ist zu {name: 'someName'}.
- Object, dessen Attribute sein können:
  - name: String.
  - type: String, unterstützt:
    - number
    - float, welches Float64Array ist
    - int, welches Int32Array ist
    - ordinal, diskreter Wert, der meistens einen String darstellt.
    - time, Zeitwert, siehe data für das Format von Zeitwerten.
  - displayName: string, im Allgemeinen im Tooltip zur Anzeige der Dimension verwendet. Falls nicht spezifiziert, wird standardmäßig name verwendet.

Wenn dimensions spezifiziert ist, wird tooltip vertikal angezeigt, wodurch Dimensionsnamen besser dargestellt werden können. Ansonsten wird tooltip horizontal angezeigt.

### 6.19.304 chart.series.i[[]].downplay.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'white' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.305 chart.series.i[].downplay.itemStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.306 chart.series.i[].downplay.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.19.307 chart.series.i[].downplay.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```



### 6.19.308 chart.series.i[].downplay.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

#### Stil der Ringdiagramm-Sektoren

Der Stil kann gesetzt werden in `series.itemStyle` für alle Sektoren dieser Serie, in `series.levels.itemStyle` für die Sektoren einer Ebene oder in `series.data.itemStyle` für einen einzelnen Sektor. Wenn `series.data.itemStyle` definiert ist, überdeckt es die Definitionen von `series.itemStyle` und `series.levels.itemStyle`.

**Priorität: `series.data.itemStyle` > `series.levels.itemStyle` > `series.itemStyle`.**

In ECharts wird *emphasis* für die Stile von Objekten verwendet, auf die die Maus zeigt. Für Ringdiagramme gibt es zwei zusätzliche States: *highlight* um Elemente hervorzuheben, die zu dem ausgewählten gehören, und *downplay* für nicht zugehörige.

### 6.19.309 chart.series.i[].downplay.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.310 chart.series.i[].downplay.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.311 chart.series.i[.].downplay.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.312 chart.series.i[.].downplay.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.313 chart.series.i[.].downplay.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.314 chart.series.i[.].downplay.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Ausrichtung des Textes. Man bemerke, dass 'left' für die innere Seite und 'right' für die äußere Seite steht.

#### Erlaubte Werte

- left
- center
- right

### 6.19.315 chart.series.i[.].downplay.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.316 chart.series.i[].downplay.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 6.19.317 chart.series.i[].downplay.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 6.19.318 chart.series.i[].downplay.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 6.19.319 chart.series.i[].downplay.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.320 chart.series.i[].downplay.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.321 chart.series.i[].downplay.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.322 chart.series.i[].downplay.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.323 chart.series.i[.].downplay.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.324 chart.series.i[.].downplay.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.325 chart.series.i[].downplay.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```



wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.326 chart.series.i[].downplay.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.327 chart.series.i[].downplay.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.328 chart.series.i[].downplay.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 6.19.329 chart.series.i[].downplay.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

label setzt den Textstil für alle Sektoren.

**Priorität: series.data.label > series.levels.label > series.label**

Textbeschriftung der Sektoren, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 6.19.330 chart.series.i[].downplay.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.331 chart.series.i[].downplay.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.332 chart.series.i[].downplay.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.333 chart.series.i[].downplay.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.19.334 chart.series.i[.].downplay.label.rotate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'radial' ]

Ist der Wert eine Zahl, steht er für eine Rotation, von -90 bis 90 Grad, wobei positive Werte gegen den Uhrzeigersinn rotieren.

Außerdem kann der String 'radial', für radielle Rotation, oder 'tangential', für tangentielle Rotation, verwendet werden.

Der Wert ist standardmäßig 'radial'. Ist keine Rotation erwünscht, kann er auf 0 gesetzt werden.

#### Erlaubte Werte

- {integer}
- radial
- tangential

### 6.19.335 chart.series.i[.].downplay.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.336 chart.series.i[.].downplay.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.337 chart.series.i[.].downplay.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.338 chart.series.i[.].downplay.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.339 chart.series.i[.].downplay.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 6.19.340 chart.series.i[.].downplay.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.341 chart.series.i[.].downplay.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.



### 6.19.342 chart.series.i[].downplay.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.343 chart.series.i[].downplay.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.344 chart.series.i[].downplay.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.345 chart.series.i[].downplay.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.346 chart.series.i[].downplay.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 6.19.347 chart.series.i[.].downplay.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.348 chart.series.i[.].downplay.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Elemente wenn die Maus auf ein Elemnt zeigt, zu dem sie keine Beziehung haben. Siehe highlightPolicy

### 6.19.349 chart.series.i[].draggable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob Knoten gezogen werden können.

### 6.19.350 chart.series.i[].drillDownIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '?' ]

Markierung wenn in einen Knoten reingebohrt werden kann.

### 6.19.351 chart.series.i[].edgeLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.352 chart.series.i[.].edgeLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.353 chart.series.i[].edgeLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.354 chart.series.i[].edgeLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.355 chart.series.i[].edgeLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.356 chart.series.i[].edgeLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.19.357 chart.series.i[].edgeLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.358 chart.series.i[].edgeLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.359 chart.series.i[.edgeLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.19.360 chart.series.i[.edgeLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke



Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.361 chart.series.i[.edgeLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.362 chart.series.i[].edgeLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.363 chart.series.i[].edgeLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.364 chart.series.i[.].edgeLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Kantenbeschriftungen.

### 6.19.365 chart.series.i[.].edgeLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.366 chart.series.i[.].edgeLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

## 6.19.367 chart.series.i[.].edgeLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.368 chart.series.i[.edgeLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.369 chart.series.i[].edgeLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.370 chart.series.i[].edgeLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.19.371 chart.series.i[].edgeLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.372 chart.series.i[].edgeLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Kantenbeschriftung angezeigt werden soll.

### 6.19.373 chart.series.i[].edgeLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.374 chart.series.i[].edgeLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.375 chart.series.i[].edgeLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.376 chart.series.i[].edgeLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.377 chart.series.i[].edgeLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.378 `chart.series.i[].edgeLabel.textShadowOffsetX`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.379 `chart.series.i[].edgeLabel.verticalAlign`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

## Erlaubte Werte

- top
- middle
- bottom

### 6.19.380 chart.series.i[.edgeLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.381 chart.series.i[.edges

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Alias von links.

### 6.19.382 chart.series.i[.].edgeSymbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['none', 'none'] ]

Symbole der beiden Enden der Kantenlinien.

Zum Beispiel:

```
edgeSymbol: ['circle', 'arrow']
```

#### Erlaubte Werte

- none
- circle
- arrow
- rect
- roundRect
- triangle
- diamond
- pin

### 6.19.383 chart.series.i[.].edgeSymbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe der Symbole an den Enden der Kantenlinien. Akzeptiert ein Array oder eine einzelne Zahl.

Zum Beispiel:

```
// Startsymbol hat Größe 5 und Endsymbol hat Größe 10
symbolSize: [5, 10],
// Beide haben Größe 10
symbolSize: 10
```

### 6.19.384 chart.series.i[].effect.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Farbe des Spezialeffekt-Symbols, welche standardmäßig identisch zu `lineStyle.color` ist.

### 6.19.385 chart.series.i[].effect.constantSpeed

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Falls die Bewegung des Spezialeffekt-Symbols eine konstante Geschwindigkeit hat, welche in Pixel pro Sekunde angegeben wird. Ist constantSpeed größer als 0, so wird period ignoriert.

### 6.19.386 chart.series.i[].effect.delay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verzögerung der Effektanimation. Kann Nummer oder Callback-Funktion sein.

### 6.19.387 chart.series.i[].effect.loop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Animation wiederholt werden soll.

### 6.19.388 chart.series.i[].effect.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Die Einstellungen für Spezialeffekte der Linien.

**Tips:** Alle Graphen mit Spureneffekten sollten auf eine individuelle Ebene gelegt werden, was bedeutet, dass ihr zlevel-Wert anders als der von andern sein muss. Die Animation dieser Ebene sollte währenddessen ausgeschaltet werden (animation: false). Ansonsten würden andere grafische Elemente in anderen Serien und die Animationsbeschriftung unnötige Geister produzieren.

## 6.19.389 chart.series.i[].effect.period

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 4 ]

Die Dauer des Spezialeffekts, in Sekunden angegeben.

## 6.19.390 chart.series.i[].effect.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Spezialeffekte angezeigt werden sollen.

## 6.19.391 chart.series.i[].effect.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'circle' ]

Das Symbol des Spezialeffekts.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

**Tip:** Der Winkel des Symbols ändert sich wenn die Tangente des Pfades sich ändert. Wird ein eigendefinierter Pfad verwendet, sollte sichergestellt werden, dass der Pfad nach oben ausgerichtet ist. So wird sichergestellt, dass das Symbol immer richtig ausgerichtet ist, wenn es sich auf dem Pfad bewegt.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

#### 6.19.392 chart.series.i[.effect.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Die Größe des Spezialeffekt-Symbols, welche als einzelne Zahl, wie 10, gesetzt werden kann. Außerdem kann ein Array verwendet werden, um Breite und Höhe getrennt zu definieren. um Beispiel, [20, 10] steht für eine Breite von 20 und eine Höhe von 10.

### 6.19.393 chart.series.i[].effect.trailLength

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.2 ]

Die Länge der Spur des Spezialeffekts. Akzeptiert werden Werte von 0 bis 1. Die Spur wird länger je größer der Wert ist.

### 6.19.394 chart.series.i[].effectType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'ripple' ]

Effekttyp. Momentan wird nur der Kräuselungseffekt 'ripple' unterstützt.

Erlaubte Werte

- ripple

### 6.19.395 chart.series.i[].emphasis.edgeLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.396 chart.series.i[].emphasis.edgeLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.397 chart.series.i[].emphasis.edgeLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.398 chart.series.i[].emphasis.edgeLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.399 chart.series.i[].emphasis.edgeLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.400 chart.series.i[].emphasis.edgeLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.19.401 chart.series.i[].emphasis.edgeLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.402 chart.series.i[].emphasis.edgeLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.403 chart.series.i[].emphasis.edgeLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.404 chart.series.i[].emphasis.edgeLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.405 chart.series.i[.emphasis.edgeLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 6.19.406 chart.series.i[].emphasis.edgeLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.407 chart.series.i[].emphasis.edgeLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.408 chart.series.i[].emphasis.edgeLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der hervorgehobenen Kante.

### 6.19.409 chart.series.i[].emphasis.edgeLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.410 chart.series.i[].emphasis.edgeLabel.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

### 6.19.411 chart.series.i[].emphasis.edgeLabel.rich.object

<b>Datentyp</b>	OBJECT
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

Label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.412 chart.series.i[.emphasis.edgeLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.413 chart.series.i[].emphasis.edgeLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.414 chart.series.i[].emphasis.edgeLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.415 chart.series.i[].emphasis.edgeLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.416 chart.series.i[].emphasis.edgeLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Kantenbeschriftung angezeigt werden soll.

### 6.19.417 chart.series.i[].emphasis.edgeLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.418 chart.series.i[].emphasis.edgeLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.419 chart.series.i[].emphasis.edgeLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.420 chart.series.i[].emphasis.edgeLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.421 chart.series.i[].emphasis.edgeLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.422 chart.series.i[].emphasis.edgeLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.423 chart.series.i[].emphasis.edgeLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

#### 6.19.424 chart.series.i[].emphasis.edgeLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes

Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.425 chart.series.i[].emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.426 chart.series.i[].emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.427 chart.series.i[].emphasis.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.428 `chart.series.i[[]].emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.429 chart.series.i[.].emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Knoten.

### 6.19.430 chart.series.i[].emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.431 chart.series.i[].emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.432 chart.series.i[].emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.433 chart.series.i[].emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.434 chart.series.i[].emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.435 chart.series.i[].emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.436 chart.series.i[].emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.437 chart.series.i[].emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.438 chart.series.i[].emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.439 chart.series.i[].emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.440 chart.series.i[].emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.441 chart.series.i[].emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.442 chart.series.i[].emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.443 chart.series.i[].emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.444 chart.series.i[].emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.19.445 chart.series.i[].emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal



- bold
- bolder
- lighter
- {integer}

### 6.19.446 chart.series.i[].emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

#### 6.19.447 chart.series.i[].emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### 6.19.448 chart.series.i[].emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

#### 6.19.449 chart.series.i[].emphasis.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

#### 6.19.450 chart.series.i[].emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des hervorgehobenen Knotens.

### 6.19.451 chart.series.i[].emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.452 chart.series.i[].emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.453 chart.series.i[].emphasis.label.position

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

#### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

#### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.454 chart.series.i[].emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.19.455 chart.series.i[].emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.456 chart.series.i[].emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.457 chart.series.i[].emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.458 chart.series.i[].emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.459 chart.series.i[].emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.460 chart.series.i[].emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.461 chart.series.i[].emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

#### 6.19.462 chart.series.i[].emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

#### 6.19.463 chart.series.i[].emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.464 chart.series.i[].emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.465 chart.series.i[].emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.466 chart.series.i[].emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.467 chart.series.i[].emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

## 6.19.468 chart.series.i[].emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.469 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.470 chart.series.i[.].emphasis.labelLine.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der visuellen Hilfslinie der Beschriftung. Steht label.position auf 'left' oder 'right', wird die Hilfslinie angezeigt.

### 6.19.471 chart.series.i[].emphasis.labelLine.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.472 chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.473 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowColor`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.474 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetX`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.475 `chart.series.i[].emphasis.labelLine.emphasis.lineStyle.shadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.476 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.477 chart.series.i[.emphasis.labelLine.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.478 chart.series.i[.emphasis.labelLine.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie im hervorgehobenen Status.

### 6.19.479 `chart.series.i[].emphasis.labelLine.emphasis.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die visuelle Hilfslinie im hervorgehobenen Zustand angezeigt werden soll.

### 6.19.480 `chart.series.i[].emphasis.labelLine.length`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des ersten Teils der hervorgehobenen visuellen Hilfslinie.

### 6.19.481 `chart.series.i[].emphasis.labelLine.lineStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.482 chart.series.i[.].emphasis.labelLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Linienstil der visuellen Hilfslinie.

### 6.19.483 chart.series.i[].emphasis.labelLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.484 chart.series.i[].emphasis.labelLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.485 chart.series.i[].emphasis.labelLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.486 chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.487 chart.series.i[].emphasis.labelLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.488 chart.series.i[.emphasis.labelLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.489 chart.series.i[.emphasis.labelLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.490 chart.series.i[.emphasis.labelLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen visuellen Hilfslinie.

### 6.19.491 `chart.series.i[].emphasis.labelLine.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die hervorgehobene visuelle Hilfslinie angezeigt werden soll.

### 6.19.492 `chart.series.i[].emphasis.lineStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel `'rgb(128, 128, 128)'`. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel `'rgba(128, 128, 128, 0.5)'`. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel `'#ccc'`. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.493 chart.series.i[.].emphasis.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



**Schlagworte / Tags**

[ Default-Wert: 0.5 ]

Die Krümmung der Kanten im Graphen.

**6.19.494 chart.series.i[].emphasis.lineStyle.object**

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

**6.19.495 chart.series.i[].emphasis.lineStyle.shadowBlur**

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.496 chart.series.i[].emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.497 chart.series.i[].emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.498 chart.series.i[].emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.499 chart.series.i[].emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.19.500 chart.series.i[].emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Elemente.

### 6.19.501 chart.series.i[].emphasis.upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.502 chart.series.i[].emphasis.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.503 chart.series.i[].emphasis.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.504 chart.series.i[].emphasis.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.505 chart.series.i[].emphasis.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.506 chart.series.i[.emphasis.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.507 chart.series.i[.emphasis.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 6.19.508 chart.series.i[].emphasis.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 6.19.509 chart.series.i[].emphasis.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.510 chart.series.i[].emphasis.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.511 chart.series.i[.emphasis.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.19.512 chart.series.i[.emphasis.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'



- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.513 chart.series.i[].emphasis.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.514 chart.series.i[].emphasis.upperLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.515 chart.series.i[].emphasis.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.516 chart.series.i[].emphasis.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen oberen Beschriftung.

### 6.19.517 chart.series.i[].emphasis.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 6.19.518 chart.series.i[].emphasis.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.519 chart.series.i[].emphasis.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.520 chart.series.i[.emphasis.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.521 chart.series.i[.emphasis.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.522 chart.series.i[].emphasis.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.523 chart.series.i[].emphasis.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.524 chart.series.i[].emphasis.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.525 chart.series.i[].emphasis.upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.526 chart.series.i[].emphasis.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.527 chart.series.i[].emphasis.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.528 chart.series.i[].emphasis.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.529 chart.series.i[].emphasis.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.530 chart.series.i[].emphasis.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.531 chart.series.i[].emphasis.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.532 chart.series.i[].emphasis.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.533 chart.series.i[].emphasis.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.534 chart.series.i[].emphasis.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.535 chart.series.i[.].encode

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert für jedes Element von data wohin es kodiert wird. Zum Beispiel:

```
option = {
  dataset: {
    source: [
      // Jede Säule ist eine Dimension.
      // Es gibt fünf Dimensionen: 0, 1, 2, 3, 4.
      [12, 44, 55, 66, 2],
      [23, 6, 16, 23, 1],
      ...
    ]
  },
  series: {
    type: 'xxx',
    encode: {
      x: [3, 1, 5],      // Dimension 3, 1, 5 ist auf X-Achse gemappt.
      y: 2,             // Dimension 2 ist auf Y-Achse gemappt.
      tooltip: [3, 2, 4] // Dimension 3, 2, 4 werden im Tooltip angezeigt.
    }
  }
}
```

Die Attribute von encode ändern sich je nach Typ des Koordinatensystems. Für cartesian2d können x und y definiert werden. Für polar können radius und angle definiert werden. Für geo können lng und lat definiert werden. Die Attribute tooltip und itemName (Name des Datenelements im tooltip) können immer definiert werden.

Wenn `dimensions` benutzt wird um Namen für bestimmte Dimensionen zu definieren, kann `encode` diese Namen direkt referenzieren. Zum Beispiel:

```
series: {
  type: 'xxx',
  dimensions: ['date', 'open', 'close', 'highest', 'lowest'],
  encode: {
    x: 'date',
    y: ['open', 'close', 'highest', 'lowest']
  }
}
```

Im besonderen kann in benutzerdefinierten Serien ein Attribut in `encode`, welches zu einer Achse gehört, auf null gesetzt werden, damit die Serie nicht von dieser Achse kontrolliert wird. Die Serientaten werden also nicht für die Länge der Achse betrachtet, und der `dataZoom` auf der Achse wird die Serie nicht filtern.

```
var option = {
  xAxis: {},
  yAxis: {},
  dataZoom: [{
    xAxisIndex: 0
  }, {
    yAxisIndex: 0
  }],
  series: {
    type: 'custom',
    renderItem: function (params, api) {
      return {
        type: 'circle',
        shape: {
          cx: 100, // x position ist immer 100
          cy: api.coord([0, api.value(0)])[1],
          r: 30
        },
        style: {
          fill: 'blue'
        }
      }
    };
  },
  encode: {
    // Dann wird die Serie nicht von der X-Achse
    // und dem zugehörigen dataZoom kontrolliert.
    x: -1,
    y: 1
  },
  data: [ ... ]
};
```

## 6.19.536 chart.series.i[].endAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: -45 ]

Der Endwinkel des Messdiagramms.

## 6.19.537 chart.series.i[].expandAndCollapse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Ein- und Ausklappen von Unterbäumen aktiviert ist, standardmäßig true. Da die Zeichenfläche limitiert ist, und ein Baum oft viele Knoten hat, können diese sich überlappen. Um dies zu vermeiden, können momentan unwichtige Unterbäume eingeklappt werden, bis sie wieder gebraucht werden.

**Hinweis: Wird ein nutzerdefiniertes Bild als Symbol eines Knotens definiert, ist es nicht möglich zwischen dem ein- und ausgeklappten Zustand durch die Füllfarbe zu unterscheiden. Das Hochladen von zwei Bildern, die die jeweiligen Zustände repräsentieren, wird momentan nicht unterstützt. Soll der Status des Knotens ersichtlich sein, wird also die Verwendung von regulären ECharts Tag-Typen, wie emptyCircle, empfohlen.**

## 6.19.538 chart.series.i[].focusNodeAdjacency

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ermöglicht dass, wenn die Maus auf eine Kante oder einen Knoten zeigt, auch mit diesem Element verbundene Kanten und Knoten hervorgehoben werden. Standardmäßig deaktiviert, kann manuell konfiguriert werden.

Mögliche Werte:

- false: Beim Zeigen auf eine Kante oder einen Knoten wird nur dieses Element hervorgehoben.
- true: Identisch zu 'allEdges'.
- 'allEdges': Beim Zeigen auf einen Knoten werden alle umgebenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.
- 'outEdges': Beim Zeigen auf einen Knoten werden alle ausgehenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.
- 'inEdges': Beim Zeigen auf einen Knoten werden alle eingehenden Kanten und deren zugehörige Knoten hervorgehoben. Beim Zeigen auf eine Kante werden alle umgebenden Knoten hervorgehoben.

Erlaubte Werte

- {boolean}
- allEdges
- outEdges
- inEdges

### 6.19.539 chart.series.i[.].force.edgeLength

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Die Distanz zwischen zwei Knoten, die durch eine Kante verbunden sind. Diese Distanz wird auch durch repulsion beeinflusst.

Dieser Wert kann ein Array sein, um eine Spannweite an Kantenlängen darzustellen. In diesem Fall werden Kanten mit größeren Werten kürzer, wodurch Knoten näher aneinander stehen, und Kanten mit kleineren Werten werden länger.

### 6.19.540 chart.series.i[].force.gravity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.1 ]

Der Schwerkraft-Faktor, der Knoten näher zum Mittelpunkt zieht. Die Knoten werden näher zum Mittelpunkt gezogen je größer der Wert ist.

### 6.19.541 chart.series.i[].force.initLayout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das initiale Layout vor dem kräftebestimmten Layout, welches das Ergebnis des kräftebestimmten Layouts beeinflusst.

Setzt standardmäßig kein eigenes Layout und verwendet die Werte x und y der Knoten für deren Positionen. Existieren diese Werte nicht, wird die Position zufällig gesetzt.

Es kann auch das zirkuläre Layout 'circular' verwendet werden.

### 6.19.542 chart.series.i[].force.layoutAnimation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Da das kräftebestimmte Layout erst nach mehreren Iterationen stabil ist, entscheidet dieser Parameter, ob die animierte Iteration des Layouts angezeigt werden soll. Es wird nicht empfohlen die Option in Browsern zu verwenden wenn es eine hohe Anzahl Knoten gibt (>100), da der Layout-Prozess den Browser zum Hängen bringen kann.

### 6.19.543 chart.series.i[].force.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen für kräftebestimmtes Layout. Das kräftebestimmte Layout simuliert ein Feder/Ladung-Modell, welches eine Abstoßung zwischen allen Knoten und eine Anziehung zwischen durch Kanten verbundenen Knoten definiert. Zuerst bewegen sich die Knoten in jeder Iteration unter den Einflüssen von Abstoßung und Anziehung. Nach mehreren Iterationen werden die Knoten in einer balancierten Position zur Ruhe kommen. So lässt sich das Modell des Graphens mit der niedrigsten lokalen Energie realisieren.

Das Ergebnis vom kräftebestimmten Layout hat gute Symmetrien und Häufungen, was auch ästhetisch ansprechend ist.

### 6.19.544 chart.series.i[].force.repulsion

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Der Abstoßungsfaktor zwischen Knoten. Die Abstoßung wird stärker und die Distanz zwischen 2 Knoten wird länger je größer dieser Wert wird.

Der Wert kann ein Array sein, um eine Spannweite an Abstoßungen darzustellen. In diesem Fall stehen höhere Werte für stärkere Abstoßung und niedrigere Werte für schwächere Abstoßung.

### 6.19.545 chart.series.i[].funnelAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Horizontale Ausrichtung. Standardmäßig zentral ausgerichtet.

Erlaubte Werte

- left
- right
- center

### 6.19.546 chart.series.i[].gap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Freiraum zwischen den Trapezen.

### 6.19.547 chart.series.i[].geoIndex

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index des geographischen Koordinatensystems mit dem dieses Diagramm kombiniert werden soll, nützlich für mehrere geografische Achsen in einem Diagramm.

### 6.19.548 chart.series.i[.].height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Höhe der Komponente.

### 6.19.549 chart.series.i[.].highlight.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'white' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.550 chart.series.i[].highlight.itemStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.551 chart.series.i[].highlight.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.19.552 chart.series.i[].highlight.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.553 chart.series.i[].highlight.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Ringdiagramm-Sektoren.

Der Stil kann in `series.itemStyle` für Sektoren dieser Serie gesetzt werden, in `series.levels.itemStyle` für die Sektoren einer Ebene, und in `series.data.itemStyle` für einen einzelnen Sektor. Wenn `series.data.itemStyle` definiert ist, überdeckt es die Einstellungen von `series.itemStyle` und `series.levels.itemStyle`.

**Priorität: `series.data.itemStyle` > `series.levels.itemStyle` > `series.itemStyle`.**

In ECharts wird *emphasis* für Stile benutzt, wenn die Maus auf das Element zeigt. Bei Ringdiagrammen gibt es zwei zusätzliche States: *highlight*, um Elemente hervorzuheben, die mit dem ausgewählten in Verbindung stehen, und *downplay*, für Elemente, die nicht mit dem ausgewählten in Verbindung stehen.

### 6.19.554 chart.series.i[].highlight.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.555 chart.series.i[].highlight.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.556 chart.series.i[].highlight.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.557 chart.series.i[].highlight.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.558 chart.series.i[].highlight.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.559 chart.series.i[].highlight.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'center' ]

Ausrichtung des Textes. Man bemerke, dass 'left' für die innere Seite und 'right' für die äußere Seite steht.

#### Erlaubte Werte

- left
- center
- right

### 6.19.560 chart.series.i[].highlight.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.561 chart.series.i[].highlight.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.562 chart.series.i[].highlight.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.563 chart.series.i[].highlight.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.564 chart.series.i[].highlight.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.565 chart.series.i[].highlight.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.566 chart.series.i[].highlight.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.567 chart.series.i[].highlight.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.568 chart.series.i[].highlight.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.19.569 chart.series.i[].highlight.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.570 chart.series.i[].highlight.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Dataset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.571 chart.series.i[].highlight.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.572 chart.series.i[].highlight.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.573 chart.series.i[].highlight.label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 6.19.574 chart.series.i[].highlight.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

label setzt den Textstil für alle Sektoren.

**Priorität: series.data.label > series.levels.label > series.label**

Textbeschriftung der Sektoren, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 6.19.575 chart.series.i[].highlight.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.576 chart.series.i[].highlight.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.



## 6.19.577 chart.series.i[].highlight.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte  
position: [10, 10],  
// Relative Prozentwerte  
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

## 6.19.578 chart.series.i[].highlight.label.rich.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.579 chart.series.i[].highlight.label.rotate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'radial' ]

Ist der Wert eine Zahl, steht er für eine Rotation, von -90 bis 90 Grad, wobei positive Werte gegen den Uhrzeigersinn rotieren.

Außerdem kann der String 'radial', für radielle Rotation, oder 'tangential', für tangentielle Rotation, verwendet werden.

Der Wert ist standardmäßig 'radial'. Ist keine Rotation erwünscht, kann er auf 0 gesetzt werden.

#### Erlaubte Werte

- {integer}
- radial
- tangential

### 6.19.580 chart.series.i[.].highlight.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.581 chart.series.i[.].highlight.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.582 chart.series.i[].highlight.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.583 chart.series.i[].highlight.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.584 chart.series.i[].highlight.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Label angezeigt werden soll.

### 6.19.585 chart.series.i[.highlight.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.586 chart.series.i[.highlight.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.19.587 chart.series.i[].highlight.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 6.19.588 chart.series.i[].highlight.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.19.589 chart.series.i[].highlight.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.590 chart.series.i[].highlight.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.591 chart.series.i[].highlight.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 6.19.592 chart.series.i[.highlight.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.19.593 chart.series.i[.highlight.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Elementstil wenn die Maus auf in Verbindung stehende Objekte zeigt. Siehe highlightPolicy.



### 6.19.594 chart.series.i[].highlightPolicy

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'descendant' ]

Wenn die Maus auf einen Sektor zeigt, erhält dieser Sektor den *emphasis*-Stil. **Ist highlightPolicy auf 'descendant' gesetzt, erhält dieser Sektor und seine Nachfahren den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'ancestor' gesetzt, erhält dieser Sektor und seine Vorfahren den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'self' gesetzt, erhält der Sektor den highlight-Stil und andere den downplay-Stil. Ist der Wert auf 'none' gesetzt, erhalten andere Sektoren nicht den downplay-Stil.**

Es wird dispatchAction verwendet um bestimmte Sektoren hervorzuheben. Der ausgewählte Sektor verwendet den Stil *emphasis*, nach der gewählten highlightPolicy zusammenhängende Sektoren verwenden den Stil *highlight*, und andere verwenden den Stil *downplay*.

```

itemStyle: {
  color: 'yellow',
  borderWidth: 2,
  emphasis: {
    color: 'red'
  },
  highlight: {
    color: 'orange'
  },
  downplay: {
    color: '#ccc'
  }
}

```

#### Erlaubte Werte

- descendant
- ancestor
- self
- none

### 6.19.595 chart.series.i[].hoverAnimation

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Animationseffekt aktiviert werden soll wenn die Maus auf die Komponente zeigt.

### 6.19.596 chart.series.i[.].hoverAnimation.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.19.597 chart.series.i[.].hoverAnimation.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Verzögerung bevor die Animation beginnt. Eine Callback-Funktion kann verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
  return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
  return (params.count - 1 - params.index) * 30;
}
```

### 6.19.598 chart.series.i[].hoverAnimation.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Spezifiziert die Verzögerung bevor die Animation aktualisiert wird. Eine Callback-Funktion kann verwendet werden, um verschiedene Verzögerungen für verschiedene Elemente zu definieren.

Zum Beispiel:

```
animationDelay: function (dataIndex, params) {
  return params.index * 30;
}
// Oder umgekehrt:
animationDelay: function (dataIndex, params) {
  return (params.count - 1 - params.index) * 30;
}
```

### 6.19.599 chart.series.i[].hoverAnimation.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.600 chart.series.i[.].hoverAnimation.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.601 chart.series.i[.].hoverAnimation.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.602 chart.series.i[.].hoverAnimation.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.603 chart.series.i[].hoverAnimation.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.19.604 chart.series.i[.].hoverAnimation.object

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Animation beim Draufzeigen aktiviert sein soll.

Dieses Attribut kann auf der Wurzelebene einer Serie gesetzt werden, wodurch alle Datenelemente der Serie davon beeinflusst werden. Das Attribut kann auch bei jedem Eintrag in series-pictorialBar.data gesetzt werden, wodurch nur dieser Eintrag von dem Attribut beeinflusst wird.

Zum Beispiel:

```

series: [{
  hoverAnimation: ... // Beeinflusst alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    hoverAnimation: ... // Beeinflusst nur dieses Datenelement.
  }, {
    value: 56
    hoverAnimation: ... // Beeinflusst nur dieses Datenelement.
  }]
}]

```

### 6.19.605 chart.series.i[.].hoverOffset

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Offset-Distanz des Sektors auf den gezeigt wird.

### 6.19.606 chart.series.i[.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.19.607 chart.series.i[.inactiveOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.05 ]

Bei der Verwendung von Brush-Auswahl werden nicht-ausgewählte Linien auf diese Transparenzrate gesetzt (was diese verdunkeln könnte).



### 6.19.608 chart.series.i[].initialTreeDepth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Die anfängliche Ebene (Tiefe) des Baums. Der Wurzelknoten ist Ebene 0, dann Ebene 1, 2, ..., bis zu den Blätterknoten. Diese Einstellung wird hauptsächlich zusammen mit Aus- und Einklappinteraktionen verwendet. Der Sinn ist, dass Knoten sich nicht gegenseitig überdecken sollen. Wird der Wert auf -1, null oder undefined gesetzt, werden alle Knoten ausgeklappt.

### 6.19.609 chart.series.i[].itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.610 chart.series.i[].itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

## 6.19.611 chart.series.i[.].itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 6.19.612 chart.series.i[.].itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.613 chart.series.i[[]].itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Grafikstil der Elemente. emphasis ist der Stil wenn das Element hervorgehoben ist, zum Beispiel wenn die Maus darauf zeigt oder über die Legendenverbindung.

## 6.19.614 chart.series.i[.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.19.615 chart.series.i[.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.19.616 chart.series.i[].itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 6.19.617 chart.series.i[].itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.19.618 chart.series.i[].itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.619 chart.series.i[].label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.620 chart.series.i[].label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.621 chart.series.i[].label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.622 chart.series.i[].label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.623 chart.series.i[].label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.624 chart.series.i[].label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.625 chart.series.i[].label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.626 chart.series.i[.].label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.627 chart.series.i[.].label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.628 chart.series.i[.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.629 chart.series.i[.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.630 chart.series.i[].label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.631 chart.series.i[.].label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.632 chart.series.i[.].label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.633 chart.series.i[].label.minAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Ist der Winkel eines Datenstücks kleiner als dieser Wert (in Grad), wird der Text nicht angezeigt. Dies wird verwendet um die Texte von kleinen Datenstücken zu verbergen.

### 6.19.634 chart.series.i[].label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textbeschriftung der Komponenten, um Informationen über die Daten grafischer Elemente wie Wert, Name und so weiter zu geben. In ECharts 2.x steht label unter itemStyle. In ECharts 3 ist label mit itemStyle auf einer Ebene, und hat wie itemStyle den Unterpunkt emphasis, um die Konfigurationsstruktur flacher zu machen.

### 6.19.635 chart.series.i[].label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.636 chart.series.i[].label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.637 chart.series.i[].label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.638 chart.series.i[].label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.19.639 chart.series.i[.].label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.640 chart.series.i[].label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.641 chart.series.i[].label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.642 chart.series.i[].label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.643 chart.series.i[.].label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.644 chart.series.i[.].label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.645 chart.series.i[.].label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.646 chart.series.i[].label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.647 chart.series.i[].label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.648 chart.series.i[].label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.649 chart.series.i[].label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.650 chart.series.i[].label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.651 chart.series.i[].label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

## 6.19.652 chart.series.i[].label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.653 chart.series.i[].labelLine.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.654 chart.series.i[.].labelLine.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Achsenlinie.

### 6.19.655 chart.series.i[].labelLine.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.656 chart.series.i[].labelLine.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```



### 6.19.657 chart.series.i[].labelLine.emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.658 chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.659 chart.series.i[].labelLine.emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.660 chart.series.i[.].labelLine.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.661 chart.series.i[.].labelLine.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.662 chart.series.i[.].labelLine.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie im hervorgehobenen Status.

### 6.19.663 `chart.series.i[].labelLine.emphasis.show`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die hervorgehobene visuelle Hilfslinie angezeigt werden soll.

### 6.19.664 `chart.series.i[].labelLine.length`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des ersten Segments der visuellen Hilfslinie.

### 6.19.665 `chart.series.i[].labelLine.length2`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Länge des zweiten Segments der visuellen Hilfslinie.

### 6.19.666 `chart.series.i[].labelLine.lineStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.667 chart.series.i[.].labelLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der Linie der visuellen Hilfslinie.

### 6.19.668 chart.series.i[].labelLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.669 chart.series.i[].labelLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.19.670 chart.series.i[].labelLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 6.19.671 chart.series.i[].labelLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.19.672 chart.series.i[].labelLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.673 chart.series.i[].labelLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.674 chart.series.i[].labelLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.675 chart.series.i[].labelLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der visuellen Hilfslinie. Wird angezeigt wenn label.position auf 'outside' steht.

### 6.19.676 chart.series.i[].labelLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die visuelle Hilfslinie angezeigt werden soll.

### 6.19.677 chart.series.i[].labelLine.smooth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die visuelle Hilfslinie geglättet werden soll. Standardmäßig false und kann auf true oder Werte von 0 bis 1 gesetzt werden, welche den Glättungsgrad festlegen.

### 6.19.678 chart.series.i[].large

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Optimierung für große Datenmengen aktiviert werden soll. Kann aktiviert werden, wenn die Anzahl der Daten Performanzprobleme verursacht.

Nach der Aktivierung kann `largeThreshold` verwendet werden, um die Minimalzahl Daten zu bestimmen, für die die Optimierung verwendet wird.

Ist die Optimierung jedoch aktiviert, können einzelne Datensätze nicht mehr angepasst werden.

### 6.19.679 `chart.series.i[].largeThreshold`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

Die Grenze an Datensätzen, ab der die Optimierung verwendet wird.

### 6.19.680 `chart.series.i[].layout`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Layout der Komponente.

Mögliche Werte:

Tree:

- 'orthogonal': Orthogonales Layout annehmen. Knoten können links nach rechts sein, rechts nach links sein, etc.
- 'radial': Radiales Layout annehmen. Der Wurzelknoten ist in der Mitte, die anderen Ebenen an Knoten bilden Ringe um den Wurzelknoten.

Graph:

- 'none': Kein Layout annehmen. Verwende x, y der Knoten als Positionsangaben.
- 'circular': Kreislayout annehmen.
- 'force': Kräftebestimmtes Layout annehmen. Siehe graph.force für weitere Informationen.

Other:

- 'horizontal': Horizontales Layout annehmen.
- 'vertical': Vertikales Layout annehmen.

### 6.19.681 chart.series.i[].layoutIterations

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 32 ]

Die Iterationen über dem Layout, welche dazu benutzt werden, die Anzahl an Überlappungen im Graphen durch bessere Platzierung der Knoten zu verringern.

Standardmäßig 32 Iterationen.

Der Test zeigt, dass die Iterationen über dem Layout nicht weniger als der Default-Wert sein sollten.

### 6.19.682 chart.series.i[].leafDepth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Wenn leafDepth gesetzt wird, wird das "drill down"-Feature aktiviert, was bedeutet, dass beim Klicken auf einen Knoten im Baum, dieser Knoten als neue Wurzel gesetzt wird und seine Kinder angezeigt werden.

leafDepth steht für die Anzahl der Ebenen, die maximal auf einmal angezeigt werden. Steht leafDepth also auf 1, wird nur eine Ebene angezeigt.

leafDepth ist standardmäßig null/undefined, was bedeutet, dass "drill down" deaktiviert ist.

### 6.19.683 chart.series.i[].leaves.emphasis.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.684 chart.series.i[].leaves.emphasis.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

## Erlaubte Werte

- dashed
- dotted
- solid

6.19.685 `chart.series.i[].leaves.emphasis.itemStyle.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

6.19.686 `chart.series.i[].leaves.emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.687 chart.series.i[.].leaves.emphasis.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Blätter.

### 6.19.688 chart.series.i[].leaves.emphasis.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.689 chart.series.i[].leaves.emphasis.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.690 chart.series.i[].leaves.emphasis.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.691 chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.692 chart.series.i[].leaves.emphasis.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]



Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.693 chart.series.i[].leaves.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.694 chart.series.i[].leaves.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.695 chart.series.i[].leaves.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.696 chart.series.i[].leaves.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.697 chart.series.i[].leaves.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.698 chart.series.i[].leaves.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.699 chart.series.i[].leaves.emphasis.label.distance

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.700 chart.series.i[.].leaves.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.701 chart.series.i[.].leaves.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.702 chart.series.i[.].leaves.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.19.703 chart.series.i[.].leaves.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.704 chart.series.i[].leaves.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.705 chart.series.i[].leaves.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.19.706 chart.series.i[].leaves.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Blattbeschriftungen.

### 6.19.707 chart.series.i[].leaves.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.708 chart.series.i[].leaves.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.709 chart.series.i[.].leaves.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom



- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.710 chart.series.i[.].leaves.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.711 chart.series.i[.].leaves.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.712 chart.series.i[].leaves.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.713 chart.series.i[].leaves.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.714 chart.series.i[].leaves.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.715 chart.series.i[].leaves.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.716 chart.series.i[].leaves.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.717 chart.series.i[].leaves.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.718 chart.series.i[].leaves.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.719 chart.series.i[].leaves.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.720 chart.series.i[].leaves.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.721 chart.series.i[].leaves.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.722 chart.series.i[].leaves.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.723 chart.series.i[].leaves.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.724 chart.series.i[].leaves.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.725 chart.series.i[].leaves.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der hervorgehobenen Komponenten.

### 6.19.726 chart.series.i[].leaves.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.727 chart.series.i[].leaves.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.728 chart.series.i[.].leaves.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1.5 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.729 chart.series.i[.].leaves.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
// aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
// Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
// absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
// ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
  // String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
  // 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.730 chart.series.i[].leaves.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil des Blattknotens im Baum.

### 6.19.731 chart.series.i[].leaves.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.732 chart.series.i[].leaves.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.733 chart.series.i[].leaves.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.734 chart.series.i[].leaves.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.735 chart.series.i[].leaves.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.736 chart.series.i[].leaves.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.737 chart.series.i[].leaves.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.738 chart.series.i[].leaves.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 6.19.739 chart.series.i[].leaves.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 6.19.740 chart.series.i[].leaves.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 6.19.741 chart.series.i[].leaves.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.742 chart.series.i[].leaves.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.743 chart.series.i[].leaves.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.744 chart.series.i[].leaves.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.745 chart.series.i[.].leaves.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal
- italic
- oblique

### 6.19.746 chart.series.i[.].leaves.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.747 chart.series.i[].leaves.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.748 chart.series.i[].leaves.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.749 chart.series.i[].leaves.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschreibt den Stil der zu den Blattknoten gehörenden Beschriftung.

### 6.19.750 chart.series.i[].leaves.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.751 chart.series.i[].leaves.label.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.752 chart.series.i[.].leaves.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left

- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.753 chart.series.i[.].leaves.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.754 chart.series.i[.].leaves.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.755 chart.series.i[].leaves.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.756 chart.series.i[].leaves.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.757 chart.series.i[].leaves.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.758 chart.series.i[].leaves.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.759 chart.series.i[].leaves.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.760 chart.series.i[].leaves.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.761 chart.series.i[].leaves.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.762 chart.series.i[].leaves.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.763 chart.series.i[].leaves.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.764 chart.series.i[].leaves.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.765 chart.series.i[].leaves.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.766 chart.series.i[].leaves.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 6.19.767 chart.series.i[].leaves.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.768 chart.series.i[].leaves.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen speziell für die Blattknoten. Zum Beispiel für andere Positionen der Beschriftungen von Blattknoten und inneren Knoten.

### 6.19.769 chart.series.i[].left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {integer}
- {percent}
- left
- center
- right

### 6.19.770 chart.series.i[].legendHoverLink

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Diagramm hervorgehoben werden soll, wenn die Maus auf die Legende zeigt.

### 6.19.771 chart.series.i[].levels.childrenVisibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Kinder werden nicht angezeigt wenn die Größe des Bereichs eines Knotens kleiner als dieser Wert ist (Einheit: Quadratpixel).

Durch diese Einstellung können die Details von Knoten versteckt werden, wenn der rechteckige Bereich nicht groß genug ist. Wenn Nutzer auf den Knoten hereinzoomen, werden die Kindknoten angezeigt wenn der Bereich größer als der angegebene Wert wird.

Siehe series-treemap.levels für Details über visuelles Encoding.

Tip: In treemap kann das childrenVisibleMin an mehreren Stellen auftauchen:

- In series-treemap, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.772 `chart.series.i[].levels.colorAlpha`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Transparenzraten (Alpha-Kanal) der Serie an. Akzeptiert Werte von 0 bis 1.

Der Wert kann zum Beispiel [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorAlpha` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.773 `chart.series.i[].levels.colorMappingBy`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'index' ]

Spezifiziert die Regel, nach der jeder Knoten seine Farbe aus der Farbliste erhält. Mögliche Werte:

- 'value':

Mappt `series-treemap.data.value` auf die Farben.

So zeigen die Farben der Knoten ihre Werte an.

`visualDimension` kann verwendet werden, um zu bestimmen, welche Datendimension für das visuelle Mapping verwendet wird.

- 'index':

Mappt den Index (Ordinalzahl) des Knotens auf die Farben. Dadurch bekommt in einer Ebene der erste Knoten die erste Farbe der Liste und der zweite Knoten die zweite Farbe.

So können aufeinanderfolgende Knoten durch ihre Farben unterschieden werden.

- 'id':

Mappt `series-treemap.data.id` auf die Farben.

Da die `id` verwendet wird um die Knoten zu identifizieren, behalten die Knoten so ihre ursprüngliche Farbe, wenn der Baum durch `setOption` verändert wird.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

## Erlaubte Werte

- value
- index
- id

### 6.19.774 `chart.series.i[.].levels.colorSaturation`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Zeigt die Spannweite an Sättigungswerten der Serie an. Akzeptiert Werte von 0 bis 1.

Zum Beispiel kann `colorSaturation` [0.3, 1] sein.

Siehe `series-treemap.levels` für Details über visuelles Encoding.

Tip: In `treemap` kann `colorMappingBy` an mehreren Stellen auftauchen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie spezifiziert.

- In jedem Array-Element von `series-treemap.levels`, wo es die vereinten Einstellungen der Ebenen des Baus spezifiziert.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten spezifiziert.

### 6.19.775 `chart.series.i[].levels.emphasis.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbe eines Knotens. Verwendet standardmäßig die globale Palette `option.color`.

### 6.19.776 `chart.series.i[].levels.emphasis.itemStyle.object`

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tips: In treemap kann das `ItemStyle` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.777 `chart.series.i[].levels.emphasis.label.align`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

#### 6.19.778 chart.series.i[.].levels.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.779 chart.series.i[.].levels.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.780 chart.series.i[.].levels.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.781 `chart.series.i[.].levels.emphasis.label.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.782 `chart.series.i[.].levels.emphasis.label.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.783 `chart.series.i[.].levels.emphasis.label.distance`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.784 chart.series.i[].levels.emphasis.label.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 6.19.785 chart.series.i[].levels.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.786 chart.series.i[].levels.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.787 chart.series.i[.].levels.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.788 chart.series.i[.].levels.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.789 chart.series.i[.].levels.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.790 chart.series.i[.].levels.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.791 chart.series.i[].levels.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.792 chart.series.i[].levels.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der auf der Ebene hervorgehobenen Knoten.

### 6.19.793 chart.series.i[].levels.emphasis.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.794 chart.series.i[].levels.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.795 chart.series.i[].levels.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

## Position der Beschriftung

### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

## 6.19.796 chart.series.i[.].levels.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.797 chart.series.i[.].levels.emphasis.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.798 chart.series.i[].levels.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.799 chart.series.i[].levels.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.800 chart.series.i[].levels.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.801 chart.series.i[].levels.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.802 chart.series.i[].levels.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.803 chart.series.i[].levels.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.804 chart.series.i[].levels.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.805 chart.series.i[].levels.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.806 chart.series.i[].levels.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.807 chart.series.i[.].levels.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.808 chart.series.i[.].levels.emphasis.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.809 chart.series.i[].levels.emphasis.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.810 chart.series.i[].levels.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.811 chart.series.i[].levels.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Beschriftungen der hervorgehobenen Knoten der Ebene.

### 6.19.812 chart.series.i[].levels.emphasis.upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

### 6.19.813 chart.series.i[.].levels.emphasis.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.814 chart.series.i[].levels.emphasis.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.815 chart.series.i[].levels.emphasis.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.816 chart.series.i[].levels.emphasis.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.817 chart.series.i[.].levels.emphasis.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.818 chart.series.i[.].levels.emphasis.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 6.19.819 chart.series.i[].levels.emphasis.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 6.19.820 chart.series.i[].levels.emphasis.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.821 chart.series.i[].levels.emphasis.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.822 chart.series.i[.].levels.emphasis.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.19.823 chart.series.i[.].levels.emphasis.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.824 chart.series.i[.].levels.emphasis.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.825 chart.series.i[.].levels.emphasis.upperLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.826 chart.series.i[.].levels.emphasis.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.827 chart.series.i[].levels.emphasis.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Oberbeschriftung des hervorgehobenen Knotens.

### 6.19.828 chart.series.i[].levels.emphasis.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

## 6.19.829 chart.series.i[.].levels.emphasis.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.830 chart.series.i[.].levels.emphasis.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.831 chart.series.i[.].levels.emphasis.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.832 chart.series.i[.].levels.emphasis.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.833 chart.series.i[].levels.emphasis.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.834 chart.series.i[].levels.emphasis.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.835 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.836 chart.series.i[].levels.emphasis.upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.837 chart.series.i[].levels.emphasis.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.838 chart.series.i[].levels.emphasis.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.839 chart.series.i[].levels.emphasis.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.840 chart.series.i[].levels.emphasis.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.841 chart.series.i[].levels.emphasis.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.842 chart.series.i[.].levels.emphasis.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.843 chart.series.i[.].levels.emphasis.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.



### 6.19.844 chart.series.i[].levels.emphasis.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.845 chart.series.i[].levels.emphasis.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.846 chart.series.i[].levels.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Die Rahmenfarbe und Lückenfarbe eines Knotens.

### 6.19.847 chart.series.i[].levels.itemStyle.borderColorSaturation

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbsättigung eines Rahmens oder Zwischenraums. Die Werte gehen von 0 bis 1.

Tips:

Wenn borderColorSaturation gesetzt ist wird borderColor deaktiviert und die endgültige Rahmenfarbe stattdessen dadurch berechnet, dass die Farbe dieses Knotens (die entweder explizit definiert oder vom Elternknoten geerbt ist) mit borderColorSaturation gemischt wird.

So kann ein Effekt implementiert werden: Verschiedene Sektionen haben jeweils andere Farbtöne in ihren Zwischenräumen, wodurch Nutzer sowohl Sektionen als auch Ebenen leicht unterscheiden können.

### Wie man Verwirrung beim Setzen des Knotenrahmens vermeidet

Wenn alle Rahmen/Zwischenräume auf die selbe Farbe gesetzt sind, kann Verwirrung auftreten, wenn Rechtecke mehrerer Ebenen auf einmal dargestellt werden.

Stattdessen kann man die Farbe des Zwischenraums setzen und die Sättigung dann mit `borderColorSaturation` verändern.

## 6.19.848 `chart.series.i[].levels.itemStyle.borderWidth`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite eines Knotens. Ist der Wert auf 0 gesetzt, gibt es keinen Rahmen.

Hinweis: Abstände zwischen Kindknoten werden durch `gapWidth` spezifiziert.

## 6.19.849 `chart.series.i[].levels.itemStyle.color`

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbe eines Knotens. Verwendet standardmäßig die globale Palette `option.color`.

## 6.19.850 `chart.series.i[].levels.itemStyle.colorAlpha`

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Transparenzrate eines Knotens, welche zwischen 0 und 1 liegt.

### 6.19.851 chart.series.i[.].levels.itemStyle.colorSaturation

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Farbsättigung eines Knotens. Die Werte gehen von 0 bis 1.

### 6.19.852 chart.series.i[.].levels.itemStyle.gapWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Freiraum zwischen Kindknoten.

### 6.19.853 chart.series.i[].levels.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tips: In treemap kann das ItemStyle Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.854 chart.series.i[].levels.itemStyle.strokeColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Strichfarbe jedes Rechtecks.

### 6.19.855 chart.series.i[].levels.itemStyle.strokeWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Strichbreite jedes Rechtecks.

## 6.19.856 chart.series.i[.].levels.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.857 chart.series.i[.].levels.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.858 chart.series.i[.].levels.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.859 chart.series.i[.].levels.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.860 chart.series.i[.].levels.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.861 chart.series.i[.].levels.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.



### 6.19.862 chart.series.i[.].levels.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.863 chart.series.i[.].levels.label.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 6.19.864 chart.series.i[.].levels.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.865 chart.series.i[.].levels.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.866 chart.series.i[.].levels.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### 6.19.867 chart.series.i[.].levels.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.868 chart.series.i[.].levels.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.869 chart.series.i[].levels.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.870 chart.series.i[].levels.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.871 chart.series.i[].levels.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Beschriftung der Knoten

Tips: In treemap kann das label Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.872 chart.series.i[].levels.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.873 chart.series.i[].levels.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.874 chart.series.i[].levels.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

6.19.875 chart.series.i[.].levels.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.876 chart.series.i[].levels.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.



## 6.19.877 chart.series.i[].levels.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.878 chart.series.i[].levels.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.879 chart.series.i[].levels.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.880 chart.series.i[.].levels.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.881 chart.series.i[.].levels.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.882 chart.series.i[.].levels.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.883 chart.series.i[].levels.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.884 chart.series.i[].levels.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.885 chart.series.i[].levels.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.886 chart.series.i[.].levels.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.887 chart.series.i[.].levels.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.888 chart.series.i[].levels.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.889 chart.series.i[].levels.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.19.890 chart.series.i[.].levels.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">TESTED</span>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

### Konfiguration auf mehreren Ebenen

treemap verwendet eine vierstufige Konfiguration:

"jeder Knoten" --> "jede Ebene" --> "jede Serie".

Wir können also jeden Knoten konfigurieren, jede Ebene des Baums konfigurieren, oder Konfigurationen für die gesamte Serie setzen. Die höchste Priorität hat die Konfiguration des Knotens.

levels ist die Konfiguration der einzelnen Ebenen, was am am meisten verwendet wird.

Zum Beispiel:

```
// Bemerke dass die Datenstruktur kein Baum ist, sondern ein Wald
data: [
  {
    name: 'nodeA',
    children: [
      {name: 'nodeAA'},
      {name: 'nodeAB'},
    ]
  },
  {
    name: 'nodeB',
    children: [
      {name: 'nodeBA'}
    ]
  }
],
levels: [
  {...}, // Konfigurationen auf der obersten Ebene der Struktur
         // (die Ebene die die obigen 'nodeA' und 'nodeB' enthält).
  {...}, // Konfigurationen der nächsten Ebene
         // (die Ebene die die obigen 'nodeAA', 'node AB', 'nodeBA' enthält)
  {...}, // Konfigurationen der nächsten Ebene
  ...
]
```

### Die Regeln über visuelles Mapping

Wenn man eine treemap entwickelt liegt der Hauptfokus darauf, wie man "verschiedene Ebenen" und "verschiedene Kategorien der selben Ebene" visuell unterscheidbar macht, was passende Werte der Rechtecksfarben, Rahmenbreiten, Rahmenfarben und auch Farbsättigungen auf allen Ebenen benötigt.

treemap verwendet diese Regel für die visuelle Konfiguration: Jede Ebene konfiguriert seinen visuellen Wert basierend auf den Einstellungen (color, colorSaturation, borderColor, colorSaturation) auf dieser Ebene. Liegt keine explizite Konfiguration vor, wird die Konfiguration des Elternknotens geerbt.

So kann dieser Effekt konfiguriert werden: Setze eine Farbliste auf der Elternebene und setze eine Farbsättigung auf der Kindebene. Jeder Knoten auf der Elternebene kriegt dann eine Farbe aus der Farbliste, und jeder Knoten auf der Kindebene kriegt eine Farbsättigung und kombiniert sie mit der Farbe seines Elternknotens um seine endgültige Farbe zu erhalten.

### Dimensionen und "Extra visuelles Mapping"

Siehe das untere Beispiel: jedes value-Feld ist ein Array, in dem jeder Eintrag eine Dimension repräsentiert.

```
[
  {
    value: [434, 6969, 8382],
    children: [
      {
        value: [1212, 4943, 5453],
        id: 'someid-1',
        name: 'Beschreibung dieses Knotens',
        children: [...]
      },
      {
        value: [4545, 192, 439],
        id: 'someid-2',
        name: 'Beschreibung dieses Knotens',
        children: [...]
      },
      ...
    ]
  },
  {
    value: [23, 59, 12],
    children: [...]
  },
  ...
]
```

treemap wird die erste Dimension (den ersten Eintrag des Arrays) auf "area" mappen. Falls wir mehr Informationen darstellen wollen, können wir eine weitere Dimension (spezifiziert durch `series-treemap.visualDimension`) auf einen anderen visuellen Typ, wie `colorSaturation`, mappen und so weiter.

### 6.19.891 `chart.series.i[[]].levels.upperLabel.align`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `align` im rich nicht gesetzt wird, wird das `align` der nächsthöheren Ebene verwendet. Zum Beispiel:



```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.892 chart.series.i[.].levels.upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.893 chart.series.i[.].levels.upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.894 chart.series.i[.].levels.upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.895 chart.series.i[.].levels.upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.896 chart.series.i[.].levels.upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.897 chart.series.i[.].levels.upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

## 6.19.898 chart.series.i[].levels.upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

## 6.19.899 chart.series.i[].levels.upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.19.900 chart.series.i[].levels.upperLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.901 chart.series.i[.].levels.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.902 chart.series.i[.].levels.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'

- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.19.903 chart.series.i[.].levels.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

#### 6.19.904 chart.series.i[.].levels.upperLabel.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Hohe des Beschriftungsbereichs.

#### 6.19.905 chart.series.i[.].levels.upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.19.906 chart.series.i[.].levels.upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

upperLabel spezifiziert, ob die Beschriftung eines Knotens mit Kindknoten angezeigt werden soll. Wenn upperLabel.show auf true gesetzt ist, werden die Beschriftungen solcher Elternknoten angezeigt.

So wie bei series-treemap.label kann die Option upperLabel an der Wurzel von series-treemap gesetzt werden, oder in series-treemap.level, oder in jedem Eintrag von series-treemap.data.

Insbesondere spezifiziert series-treemap.label den Stil, wenn der Knoten ein Blatt ist, während series-treemap.upperLabel den Stil spezifiziert, wenn ein Knoten Kinder hat. In diesem Fall wird die Beschriftung im inneren oberen Bereich des Knotens angezeigt.

Tips: In treemap kann das upperLabel Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

## 6.19.907 chart.series.i[.].levels.upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.908 chart.series.i[.].levels.upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.909 chart.series.i[.].levels.upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

## Position der Beschriftung

### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

## Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

## 6.19.910 chart.series.i[.].levels.upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.911 chart.series.i[.].levels.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.912 chart.series.i[.].levels.upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.913 chart.series.i[.].levels.upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.914 chart.series.i[.].levels.upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.915 chart.series.i[.].levels.upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.916 chart.series.i[.].levels.upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.917 chart.series.i[.].levels.upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.918 chart.series.i[.].levels.upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.919 chart.series.i[.].levels.upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.920 chart.series.i[.].levels.upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.921 chart.series.i[.].levels.upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.922 chart.series.i[.].levels.upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.923 chart.series.i[].levels.upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.924 chart.series.i[].levels.upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.925 chart.series.i[].levels.visibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Ein Knoten wird nicht angezeigt wenn die Größe seines Bereichs kleiner als dieser Wert ist (Einheit: px quadrat).

So werden kleine Knoten, die ansonsten eng aneinander stehen würden, versteckt. Wenn ein Nutzer die treemap vergrößert, vergrößert sich auch die Bereichsgröße, und das Rechteck wird angezeigt, wenn sie über diese Grenze steigt.

Siehe serie-treemap.levels für Details zu visuellem Encoding.

Tips: In treemap kann das visibleMin Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.926 chart.series.i[].levels.visualDimension

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

treemap kann beliebige Dimensionen der Daten ins Visuelle mappen.

Der Wert von `series-treemap.data` kann ein Array sein. Jeder Eintrag des Arrays steht für eine "dimension". `visualDimension` spezifiziert die Dimension, auf der visuelles Mapping durchgeführt wird.

Siehe `series-treemap.levels` für Details zu visuellem Encoding.

Tips: In `treemap` kann das `visualDimension` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.927 `chart.series.i[.].levels.visualMax`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn `colorMappingBy` auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von `visualMin` oder `visualMax` bestimmt werden.

### 6.19.928 `chart.series.i[.].levels.visualMin`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: null ]

Der minimale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn colorMappingBy auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von visualMin oder visualMax bestimmt werden.

### 6.19.929 chart.series.i[].lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.930 chart.series.i[.].lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Linienstil

## 6.19.931 chart.series.i[].lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.19.932 chart.series.i[].lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.933 chart.series.i[].lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.934 chart.series.i[].lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.935 chart.series.i[].lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.936 chart.series.i[.].lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.937 chart.series.i[.].lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Linienbreite.

### 6.19.938 chart.series.i[.].links.emphasis.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.939 chart.series.i[].links.emphasis.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:



```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.940 chart.series.i[.links.emphasis.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.941 chart.series.i[.links.emphasis.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 6.19.942 chart.series.i[].links.emphasis.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

## 6.19.943 chart.series.i[].links.emphasis.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

## 6.19.944 chart.series.i[].links.emphasis.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.945 chart.series.i[.links.emphasis.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.946 chart.series.i[.links.emphasis.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- oblique
- italic

### 6.19.947 chart.series.i[.links.emphasis.label.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.948 chart.series.i[.].links.emphasis.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.

- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

**Beispiel:**

```
formatter: '{b}: {@score}'
```

**Callback-Funktionen**

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 6.19.949 chart.series.i[.links.emphasis.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.950 chart.series.i[.].links.emphasis.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.951 chart.series.i[.].links.emphasis.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Beschriftung des hervorgehobenen Elements.

## 6.19.952 chart.series.i[.links.emphasis.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.953 chart.series.i[.links.emphasis.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte  
position: [10, 10],  
// Relative Prozentwerte  
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.954 chart.series.i[.links.emphasis.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.955 chart.series.i[.links.emphasis.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.956 chart.series.i[].links.emphasis.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.957 chart.series.i[].links.emphasis.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.19.958 chart.series.i[].links.emphasis.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.959 chart.series.i[].links.emphasis.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.960 chart.series.i[].links.emphasis.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.961 chart.series.i[].links.emphasis.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.962 chart.series.i[].links.emphasis.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.963 chart.series.i[].links.emphasis.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.964 chart.series.i[].links.emphasis.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.965 `chart.series.i[.links.emphasis.label.textShadowOffsetY`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.966 `chart.series.i[.links.emphasis.label.verticalAlign`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im rich nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

### 6.19.967 chart.series.i[.links.emphasis.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.968 chart.series.i[.links.emphasis.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.969 chart.series.i[].links.emphasis.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Liniestil des hervorgehobenen Elements.

### 6.19.970 chart.series.i[].links.emphasis.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.971 chart.series.i[].links.emphasis.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.



Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.972 chart.series.i[].links.emphasis.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.973 chart.series.i[].links.emphasis.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.974 chart.series.i[].links.emphasis.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.975 chart.series.i[.links.emphasis.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.976 chart.series.i[.links.emphasis.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.977 chart.series.i[].links.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des hervorgehobenen Elements.

### 6.19.978 chart.series.i[].links.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.979 chart.series.i[.links.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.980 chart.series.i[.links.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.981 chart.series.i[.links.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.982 chart.series.i[.links.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.983 chart.series.i[.links.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.19.984 chart.series.i[].links.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.985 chart.series.i[].links.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.986 chart.series.i[].links.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.987 chart.series.i[.].links.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.988 chart.series.i[.links.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:



```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.989 chart.series.i[.].links.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.990 chart.series.i[.].links.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.991 chart.series.i[].links.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Elements.

### 6.19.992 chart.series.i[].links.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.993 chart.series.i[.].links.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'middle' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom

- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.994 chart.series.i[.links.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.995 chart.series.i[.].links.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.996 chart.series.i[.links.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.997 chart.series.i[.links.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

## 6.19.998 chart.series.i[.links.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.999 chart.series.i[].links.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1000 chart.series.i[].links.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.1001 chart.series.i[].links.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1002 chart.series.i[].links.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1003 chart.series.i[].links.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1004 chart.series.i[].links.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1005 chart.series.i[].links.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1006 chart.series.i[].links.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}

```

## Erlaubte Werte

- top
- middle
- bottom

### 6.19.1007 chart.series.i[.links.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1008 chart.series.i[.links.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1009 chart.series.i[].links.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Krümmung der Kante, unterstützt Werte von 0 bis 1. Die Krümmung wird stärker wenn der Wert größer wird.

### 6.19.1010 chart.series.i[].links.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Kanten.

### 6.19.1011 chart.series.i[].links.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1012 chart.series.i[].links.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1013 chart.series.i[].links.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1014 chart.series.i[].links.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1015 chart.series.i[.links.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1016 chart.series.i[.links.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

## 6.19.1017 chart.series.i[].links.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

## 6.19.1018 chart.series.i[].links.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beziehungsdaten zwischen Knoten. Beispiel:

```
links: [{
  source: 'n1',
  target: 'n2'
}, {
  source: 'n2',
  target: 'n3'
}]
```

## 6.19.1019 chart.series.i[].links.source

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Quellknotens der Kante.

### 6.19.1020 chart.series.i[].links.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Kantenenden. Kann ein Array mit zwei Elementen sein, um die Enden separat zu definieren, oder ein String, um sie zusammen zu definieren.

### 6.19.1021 chart.series.i[].links.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße der Kantenenden. Kann ein Array mit zwei Elementen sein, um die Enden separat zu definieren, oder ein String, um sie zusammen zu definieren.

### 6.19.1022 chart.series.i[].links.target

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name des Zielknotens der Kante.

### 6.19.1023 chart.series.i[.links.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kantenwert. Kann in kräftebasiertem Graphen auf die kantenlänge gemappt werden.

### 6.19.1024 chart.series.i[.markArea.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.19.1025 chart.series.i[].markArea.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
    // Verzögerung für spätere Daten ist größer  
    return idx * 100;  
}
```

### 6.19.1026 chart.series.i[].markArea.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.19.1027 chart.series.i[].markArea.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.1028 chart.series.i[].markArea.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.1029 chart.series.i[].markArea.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn

- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1030 chart.series.i[[]].markArea.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut

- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1031 chart.series.i[].markArea.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.19.1032 chart.series.i[].markArea.data.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Das Ausmaß des Bereichs wird in data festgelegt. data akzeptiert Arrays mit zwei Elementen, welche den linken oberen und den rechten unteren Punkt der rechteckigen Markierungsebene beschreiben. Jeder Eintrag kann wie folgt definiert werden:

1. Spezifiziere die Koordinaten im Koordinatensystem des Bildschirms mit x und y, wobei die Werte Pixelwerte (wie 5) oder Prozentwerte (wie '35%') sein können.
2. Spezifiziere die Koordinaten im (kartesischen) Koordinatensystem der Daten durch coord, welches auch die Werte 'min', 'max' und 'average' akzeptiert (z.B. coord: [23, 'min'] oder coord: ['average', 'max']).
3. Spezifiziere den minimalen, maximalen oder durchschnittlichen Wert von series.data durch type, wobei valueIndex oder valueDim verwendet werden können um die Dimension zu bestimmen, auf der die Werte berechnet werden.
4. In einem kartesischen System können nur xAxis oder yAxis spezifiziert werden, um einen markierten Bereich basierend auf nur der X- oder Y-Achse zu definieren

Die Priorität ist wie oben wenn mehr als eine der oberen Definitionen benutzt wird.

```
data: [  
  [  
    {  
      name: 'Vom Durchschnitt zum Maximum',  
      type: 'average'  
    },  
    {  
      type: 'max'  
    }  
  ],  
  [  
    {  
      name: 'Bereich zwischen zwei Punkten in Datenkoordinaten markieren',  
      coord: [10, 20]  
    },  
    {  
      coord: [20, 30]  
    }  
  ],  
  [  
    {  
      name: 'Von 60 bis 80',  
      yAxis: 60  
    },  
    {  
      yAxis: 80  
    }  
  ],  
  [  
    {  
      name: 'Markierter Bereich umschließt alle Daten'  
      coord: ['min', 'min']  
    },  
    {  
      coord: ['max', 'max']  
    }  
  ],  
  [  
    {  
      name: 'Bereich um zwei Bildschirmpunkte markieren',  
      x: 100,  
      y: 100  
    }, {  
      x: '90%',  
      y: '10%'  
    }  
  ]  
]
```

### 6.19.1033 chart.series.i[].markArea.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.1034 chart.series.i[].markArea.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.1035 chart.series.i[].markArea.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.1036 chart.series.i[].markArea.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1037 chart.series.i[].markArea.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.1038 chart.series.i[].markArea.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.1039 chart.series.i[].markArea.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 6.19.1040 chart.series.i[].markArea.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1041 chart.series.i[].markArea.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil des hervorgehobenen Markierungsbereichs.

### 6.19.1042 chart.series.i[].markArea.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1043 chart.series.i[].markArea.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1044 chart.series.i[].markArea.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1045 chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1046 chart.series.i[].markArea.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1047 chart.series.i[].markArea.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Markierungsbereichs.

### 6.19.1048 chart.series.i[].markArea.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1049 chart.series.i[].markArea.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1050 chart.series.i[].markArea.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1051 chart.series.i[].markArea.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1052 chart.series.i[].markArea.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1053 chart.series.i[].markArea.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.1054 chart.series.i[].markArea.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1055 chart.series.i[].markArea.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1056 chart.series.i[].markArea.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.1057 chart.series.i[].markArea.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1058 chart.series.i[].markArea.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1059 chart.series.i[].markArea.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.1060 chart.series.i[].markArea.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.1061 chart.series.i[].markArea.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 6.19.1062 chart.series.i[].markArea.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 6.19.1063 chart.series.i[].markArea.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

## 6.19.1064 chart.series.i[].markArea.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1065 chart.series.i[].markArea.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1066 chart.series.i[].markArea.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.1067 chart.series.i[].markArea.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1068 chart.series.i[].markArea.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.1069 chart.series.i[].markArea.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

## Erlaubte Werte

- normal
- italic
- oblique

## 6.19.1070 chart.series.i[].markArea.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.1071 chart.series.i[].markArea.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.19.1072 chart.series.i[].markArea.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.19.1073 chart.series.i[].markArea.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des hervorgehobenen Markierungsbereichs.

### 6.19.1074 chart.series.i[].markArea.label.emphasis.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.1075 chart.series.i[].markArea.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.1076 chart.series.i[].markArea.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.1077 chart.series.i[].markArea.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

## Schlagworte / Tags

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.1078 chart.series.i[].markArea.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.1079 chart.series.i[].markArea.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.1080 chart.series.i[].markArea.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.1081 chart.series.i[].markArea.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1082 chart.series.i[].markArea.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1083 chart.series.i[].markArea.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1084 chart.series.i[].markArea.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1085 chart.series.i[].markArea.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1086 chart.series.i[].markArea.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 6.19.1087 chart.series.i[].markArea.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.19.1088 chart.series.i[].markArea.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 6.19.1089 chart.series.i[].markArea.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1090 chart.series.i[].markArea.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.1091 chart.series.i[].markArea.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.19.1092 chart.series.i[].markArea.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.19.1093 chart.series.i[].markArea.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.1094 chart.series.i[].markArea.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.19.1095 chart.series.i[].markArea.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.1096 chart.series.i[].markArea.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1097 chart.series.i[].markArea.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.19.1098 chart.series.i[].markArea.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Markierungsbereichs.

### 6.19.1099 chart.series.i[].markArea.label.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.1100 chart.series.i[].markArea.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.1101 chart.series.i[].markArea.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom



- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.1102 chart.series.i[].markArea.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.1103 chart.series.i[].markArea.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.1104 chart.series.i[].markArea.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.1105 chart.series.i[].markArea.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.1106 chart.series.i[].markArea.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1107 chart.series.i[].markArea.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1108 chart.series.i[].markArea.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1109 chart.series.i[].markArea.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1110 chart.series.i[].markArea.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1111 chart.series.i[].markArea.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1112 chart.series.i[].markArea.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1113 chart.series.i[].markArea.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1114 chart.series.i[].markArea.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1115 chart.series.i[].markArea.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.1116 chart.series.i[].markArea.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1117 chart.series.i[].markArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wird verwendet um einen Bereich im Diagramm zu markieren. Zum Beispiel ein Zeitintervall.

### 6.19.1118 chart.series.i[].markArea.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

### 6.19.1119 chart.series.i[].markLine.animation

<b>Datentyp</b>	BOOLEAN
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.19.1120 chart.series.i[].markLine.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedenen Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
  // Verzögerung für spätere Daten ist größer  
  return idx * 100;  
}
```

### 6.19.1121 chart.series.i[].markLine.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {  
    // Verzögerung für spätere Daten ist größer  
    return idx * 100;  
}
```

### 6.19.1122 chart.series.i[].markLine.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {  
    // Dauer ist für spätere Daten länger  
    return idx * 100;  
}
```

### 6.19.1123 chart.series.i[].markLine.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.1124 chart.series.i[].markLine.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut

- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1125 chart.series.i[].markLine.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut

- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1126 chart.series.i[].markLine.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.19.1127 chart.series.i[].markLine.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagnworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 6.19.1128 chart.series.i[].markLine.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der hervorgehobenen Markierungslinie.

## 6.19.1129 chart.series.i[].markLine.label.emphasis.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'end' ]

Position der Beschriftung.

Erlaubte Werte

- start
- middle
- end

## 6.19.1130 chart.series.i[].markLine.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1131 chart.series.i[].markLine.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Seriename.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:



```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.1132 chart.series.i[].markLine.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung der Markierungslinie.

### 6.19.1133 chart.series.i[].markLine.label.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'end' ]

Position der Beschriftung.

### Erlaubte Werte

- start
- middle
- end

## 6.19.1134 chart.series.i[].markLine.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

## 6.19.1135 chart.series.i[].markLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1136 chart.series.i[].markLine.lineStyle.curveness

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Krümmung der Kante, welche Werte von 0 bis 1 unterstützt. Je größer der Wert, desto größer die Krümmung.

### 6.19.1137 chart.series.i[].markLine.lineStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #000 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1138 chart.series.i[].markLine.lineStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil der hervorgehobenen Markierungslinie.

### 6.19.1139 chart.series.i[].markLine.lineStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1140 chart.series.i[].markLine.lineStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

## 6.19.1141 chart.series.i[].markLine.lineStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

## 6.19.1142 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.19.1143 chart.series.i[].markLine.lineStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1144 chart.series.i[].markLine.lineStyle.emphasis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.1145 chart.series.i[].markLine.lineStyle.emphasis.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.1146 chart.series.i[].markLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Markierungslinie.

### 6.19.1147 `chart.series.i[].markLine.lineStyle.opacity`

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1148 `chart.series.i[].markLine.lineStyle.shadowBlur`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit `shadowColor`, `shadowOffsetX` und `shadowOffsetY` verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1149 chart.series.i[].markLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1150 chart.series.i[].markLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1151 chart.series.i[].markLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1152 chart.series.i[].markLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.1153 chart.series.i[].markLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Linienbreite.

### 6.19.1154 chart.series.i[].markLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verwende zu Illustrationszwecken eine Linie im Graphen.

### 6.19.1155 chart.series.i[].markLine.precision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Präzision des Wertes der Markierungslinie, was nützlich ist, wenn die Linie den durchschnittlichen Wert anzeigen soll.

### 6.19.1156 chart.series.i[].markLine.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

### 6.19.1157 chart.series.i[].markLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symboltyp an den beiden Enden der Markierungslinie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren. Siehe data.symbol für weitere Informationen zum Format.

### 6.19.1158 chart.series.i[].markLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße an den beiden Enden der Markierungslinie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren.

**Vorsicht:** Es können nicht Breite und Höhe getrennt gesetzt werden wie beim normalen SymbolSize.

### 6.19.1159 chart.series.i[].markPoint.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Animationen aktiviert sein sollen.

### 6.19.1160 chart.series.i[].markPoint.animationDelay

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die erste Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.19.1161 chart.series.i[].markPoint.animationDelayUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verzögerung bevor die Animation aktualisiert wird, unterstützt Callback-Funktionen, um verschiedenen Daten verschieden Animationseffekte zu geben.

Zum Beispiel:

```
animationDelay: function (idx) {
  // Verzögerung für spätere Daten ist größer
  return idx * 100;
}
```

### 6.19.1162 chart.series.i[].markPoint.animationDuration

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1000 ]

Dauer der ersten Animation, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.1163 chart.series.i[].markPoint.animationDurationUpdate

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Dauer, die die Animation zum Durchlaufen braucht, unterstützt Callback-Funktionen, um verschiedenen Daten verschiedene Animationseffekte zu geben:

```
animationDuration: function (idx) {
  // Dauer ist für spätere Daten länger
  return idx * 100;
}
```

### 6.19.1164 chart.series.i[].markPoint.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Linderungsmethode für die erste Animation.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut



- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1165 chart.series.i[.].markPoint.animationEasingUpdate

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'cubicOut' ]

Lockerungsmethode der Animation

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn
- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn

- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.19.1166 chart.series.i[].markPoint.animationThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Spezifiziert einen Schwellenwert für die Anzahl an animierten Grafiken. Ist die Anzahl der zu animierenden Grafiken größer als der Schwellenwert, wird die Animation ausgeschaltet.

### 6.19.1167 chart.series.i[].markPoint.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.1168 chart.series.i[].markPoint.itemStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.1169 chart.series.i[].markPoint.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.1170 chart.series.i[].markPoint.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte
aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding
Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter
absolute Pixelwerte.
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius,
ähnlich zum linearen Gradienten.
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
// Fill with texture
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt,
String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf
'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1171 chart.series.i[].markPoint.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.19.1172 chart.series.i[].markPoint.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- dashed
- dotted
- solid

### 6.19.1173 chart.series.i[].markPoint.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.19.1174 chart.series.i[].markPoint.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe. Wird standardmäßig aus option.color Palette bezogen.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.19.1175 chart.series.i[].markPoint.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Stil des hervorgehobenen Markierungspunktes.

### 6.19.1176 chart.series.i[].markPoint.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1177 chart.series.i[].markPoint.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```



### 6.19.1178 chart.series.i[].markPoint.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1179 chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1180 chart.series.i[].markPoint.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1181 chart.series.i[].markPoint.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Markierungspunktes.

### 6.19.1182 chart.series.i[].markPoint.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1183 chart.series.i[].markPoint.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1184 chart.series.i[].markPoint.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1185 chart.series.i[].markPoint.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1186 chart.series.i[].markPoint.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1187 chart.series.i[].markPoint.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.19.1188 chart.series.i[].markPoint.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1189 chart.series.i[].markPoint.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1190 chart.series.i[].markPoint.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.1191 chart.series.i[].markPoint.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1192 chart.series.i[].markPoint.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1193 chart.series.i[].markPoint.label.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.1194 chart.series.i[].markPoint.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.1195 chart.series.i[].markPoint.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.



### 6.19.1196 chart.series.i[].markPoint.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1197 chart.series.i[].markPoint.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.1198 chart.series.i[].markPoint.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1199 chart.series.i[].markPoint.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1200 chart.series.i[].markPoint.label.emphasis.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Distanz zum grafischen Host-Element. Funktioniert wenn position ein String-Wert ist (wie 'top', 'insideRight')

### 6.19.1201 chart.series.i[].markPoint.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1202 chart.series.i[].markPoint.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.1203 chart.series.i[].markPoint.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

## Erlaubte Werte

- normal
- italic
- oblique

## 6.19.1204 chart.series.i[].markPoint.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.1205 chart.series.i[].markPoint.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

## Schlagworte / Tags

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- `{a}`: Serienname.
- `{b}`: Name des Datenelements.
- `{c}`: Wert des Datenelements.
- `{@xxx}`: Der Wert einer Dimension namens 'xxx', so steht `{@product}` für den Wert der Dimension 'product'.
- `{@[n]}`: Der Wert der Dimension bei Index n, so steht `{@[3]}` für den Wert von `dimensions[3]`.

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei `params` das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Serienname
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

## 6.19.1206 chart.series.i[].markPoint.label.emphasis.height

### Datentyp

UNKNOWN

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1207 chart.series.i[].markPoint.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.1208 chart.series.i[].markPoint.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des hervorgehobenen Markierungspunktes.

### 6.19.1209 chart.series.i[].markPoint.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.1210 chart.series.i[].markPoint.label.emphasis.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.1211 chart.series.i[].markPoint.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:



```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.1212 chart.series.i[.].markPoint.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.1213 chart.series.i[].markPoint.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.1214 chart.series.i[].markPoint.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.1215 chart.series.i[].markPoint.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1216 chart.series.i[].markPoint.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1217 chart.series.i[].markPoint.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1218 chart.series.i[].markPoint.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1219 chart.series.i[].markPoint.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1220 chart.series.i[].markPoint.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1221 chart.series.i[].markPoint.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1222 chart.series.i[].markPoint.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1223 chart.series.i[].markPoint.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1224 chart.series.i[].markPoint.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.1225 chart.series.i[].markPoint.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1226 chart.series.i[].markPoint.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1227 chart.series.i[].markPoint.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.19.1228 chart.series.i[].markPoint.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

## 6.19.1229 chart.series.i[].markPoint.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal



- bold
- bolder
- lighter
- {integer}

### 6.19.1230 chart.series.i[[]].markPoint.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

#### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

#### Beispiel:

```
formatter: '{b}: {@score}'
```

#### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}

```

### 6.19.1231 chart.series.i[].markPoint.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1232 chart.series.i[].markPoint.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.1233 chart.series.i[].markPoint.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des Markierungspunktes.

### 6.19.1234 chart.series.i[].markPoint.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.1235 chart.series.i[].markPoint.label.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'inside' ]

Position der Beschriftung

**Kann die folgenden Werte annehmen:**

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom

- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.1236 chart.series.i[].markPoint.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.1237 chart.series.i[].markPoint.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.19.1238 chart.series.i[].markPoint.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.19.1239 chart.series.i[].markPoint.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.19.1240 chart.series.i[].markPoint.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1241 chart.series.i[].markPoint.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1242 chart.series.i[].markPoint.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1243 chart.series.i[].markPoint.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1244 chart.series.i[].markPoint.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1245 chart.series.i[].markPoint.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1246 chart.series.i[].markPoint.label.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1247 chart.series.i[].markPoint.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1248 chart.series.i[].markPoint.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.1249 chart.series.i[].markPoint.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

## Erlaubte Werte

- top
- middle
- bottom

## 6.19.1250 chart.series.i[].markPoint.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1251 chart.series.i[].markPoint.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Markierungspunkt in einem Diagramm.

### 6.19.1252 chart.series.i[].markPoint.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

### 6.19.1253 chart.series.i[].markPoint.symbol

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'pin' ]

Symbol des Markierungspunktes.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

### 6.19.1254 chart.series.i[].markPoint.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 6.19.1255 chart.series.i[].markPoint.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

#### Erlaubte Werte

- {integer}
- {percent}

### 6.19.1256 chart.series.i[].markPoint.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 6.19.1257 chart.series.i[].markPoint.symbolSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Symbolgröße. Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

Müssen die Größen der Symbole unterschiedlich sein, kann eine Callback-Funktion im folgenden Format verwendet werden:

```
(value: Array|number, params: Object) => number|Array
```

Der erste Parameter value ist der Wert in data, der zweite Parameter params enthält die restlichen Parameter vom data-Objekt.

### 6.19.1258 chart.series.i[].max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Der spezifiziert maximale Wert.

### 6.19.1259 chart.series.i[].maxOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Maximale Undurchsichtigkeit. Ist gültig für coordinateSystem vom Typ 'geo'.

### 6.19.1260 chart.series.i[].maxSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die gemappte Breite vom maximalen Datenwert max.

Der Wert kann ein absoluter Pixelwert oder ein Prozentwert relativ zur Breite des Layouts sein.

### 6.19.1261 chart.series.i[].min

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der spezifizierte Minimalwert.

### 6.19.1262 chart.series.i[].minAngle

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Minimale Winkel eines Sektors (0 ~ 360). Dieser Wert verhindert, dass Sektoren mit zu kleinen Werten so klein werden, dass die Nutzerinteraktion beeinträchtigt wird.

### 6.19.1263 chart.series.i[].minOpacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Minimale Undurchsichtigkeit. Ist gültig für coordinateSystem vom Typ 'geo'.

### 6.19.1264 chart.series.i[].minSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '100%' ]

Die gemappte Breite vom maximalen Datenwert max.

Der Wert kann ein absoluter Pixelwert oder ein Prozentwert relativ zur Breite des Layouts sein. Wenn der Graph des kleinsten Werts kein Dreieck sein soll, kann dieser Wert größer als 0 gesetzt werden.

### 6.19.1265 chart.series.i[].nameMap

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Mapping der Namen für angepasste Gebiete. Zum Beispiel:

```
{
  'China' : '??'
}
```

### 6.19.1266 chart.series.i[].nodeClick

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: treemap: 'zoomToNode', sunburst: 'rootToNode' ]

Das Verhalten wenn auf einen Knoten geklickt wird. Mögliche Werte sind:

- false: Nichts passiert.
- In treemap-Graphen: 'zoomToNode': Auf angeklickten Knoten zoomen.
- In sunburst-Graphen: 'rootToNode': Angeklickten Sektor als Wurzel benutzen
- 'link': Wenn link gesetzt ist, leitet die Seite dahin weiter.

## 6.19.1267 chart.series.i[].nodeGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Zwischenraum zwischen allen zwei Rechtecken in jeder Spalte des Graphs.

## 6.19.1268 chart.series.i[].nodes

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Alias von data.

## 6.19.1269 chart.series.i[].nodeScaleRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.6 ]

Zoomrate der Knoten wenn die Maus rein oder raus zoomt. Wenn der Wert auf 0 steht, zoomt der Knoten nicht, wenn die Maus zoomt, verändert sich also in seiner Größe nicht.

### 6.19.1270 chart.series.i[].nodeWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Die Breite der rechteckigen Knoten im Graph.

### 6.19.1271 chart.series.i[].orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Richtung des Layouts im Diagramm.

Mögliche Werte:

Tree (wenn layout 'orthogonal' ist):

- 'LR', 'horizontal': Von links nach rechts.
- 'RL': Von rechts nach links.
- 'TB', 'vertical': Von oben nach unten.
- 'BT': Von unten nach oben.

Sankey:

- 'horizontal': Von links nach rechts.
- 'vertical': Von oben nach unten.

## Erlaubte Werte

- LR
- RL
- TB
- BT
- horizontal
- vertical

### 6.19.1272 chart.series.i[].parallelIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der Parallelkoordinaten mit denen kombiniert werden soll, was nützlich für mehrere Parallelachsen in einem Diagramm ist.

### 6.19.1273 chart.series.i[].pointer.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '80%' ]

Die Länge des Zeigers, kann ein absoluter Wert oder ein Prozentwert relativ zum Radius sein.

## Erlaubte Werte

- {integer}
- {percent}

## 6.19.1274 chart.series.i[].pointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeiger des Messdiagramms.

## 6.19.1275 chart.series.i[].pointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Zeiger angezeigt werden soll.

## 6.19.1276 chart.series.i[].pointer.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Die Breite des Zeigers.

### 6.19.1277 chart.series.i[].polarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index des Polarkoordinatensystems mit dem kombiniert werden soll, was nützlich für mehrere Polarachsen in einem Diagramm ist.

### 6.19.1278 chart.series.i[].polyline

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linien als Polygonzüge gezeichnet werden sollen.

Standardmäßig false, wodurch nur eine Linie zwischen zwei Punkten gemalt werden kann.

Wird der Wert auf true gesetzt, akzeptiert data.coords mehr als zwei koordinaten, wodurch ein Polygonzug gezeichnet werden kann. Nützlich wenn Daten aus GPS-Verfolgungen visualisiert werden sollen.

### 6.19.1279 chart.series.i[].progressive

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

progressive spezifiziert die Anzahl der grafischen Elemente die in einem Frame (etwa 16ms) gerendert werden können, wenn progressives Rendering aktiviert ist.

Wenn die Anzahl der Datenpunkte von Tausend bis über 10 Millionen steigt, dauert es zu lange, alle grafischen Elemente zu rendern. Seit ECharts 4 wird progressives Rendering im Workflow unterstützt, wodurch die Daten Block für Block mit jedem Frame verarbeitet und gerendert werden. Somit wird der UI-Thread des Browsers nicht blockiert.

### 6.19.1280 chart.series.i[].progressiveChunkMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mod' ]

Ansatz zum aufteilen der Daten in Blöcke. Mögliche Werte:

- 'sequential': Teile Daten nach Datenindex auf.
- 'mod': Teile Daten nach Modulo auf, wodurch die Daten jedes Blocks aus verschiedenen Bereichen der Datenliste kommen, was einen ansprechenderen visuellen Effekt beim progressiven Rendern geben kann.

#### Erlaubte Werte

- sequential
- mod

### 6.19.1281 chart.series.i[].progressiveThreshold

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3000 ]

Wenn die momentane Anzahl DATen über dieser Grenze liegt, wird das progressive Rendering aktiviert.

### 6.19.1282 chart.series.i[].radarIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Index der Radarkomponente die das Radadiagramm benutzt.

### 6.19.1283 chart.series.i[].radius

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Radius des Diagramms.

#### Sunburst-, Kuchendiagramm:

[ Default-Wert: [0, '75%'] ]

Mögliche Werte sind:

- Zahl: Spezifiziert den äußeren Radius direkt.
- String: Zum Beispiel bedeutet '20%', dass der äußere Radius 20% der Größe des Viewports beträgt (des Kleinen zwischen der Breite und Höhe des Diagrammcontainers).

- `Array.<Zahl|String>`: Das erste Element spezifiziert den inneren Radius, das zweite Element spezifiziert den äußeren Radius. Beide folgen den obigen Definitionen.

### Messdiagramm:

[ Default-Wert: '75%' ]

Kann ein absoluter Wert oder ein Prozentwert relativ zum Kleineren der halben Breite und der halben Höhe des Containers sein.

### Erlaubte Werte

- {integer}
- {percent}

## 6.19.1284 `chart.series.i[].realtime`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert werden soll.

## 6.19.1285 `chart.series.i[].renderItem.object`

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Nutzerdefinierte Serien verlangen von Entwicklern, ihre eigene Renderlogik zu schreiben. Diese Renderlogik wird als `renderItem` bezeichnet.

Zum Beispiel:

```

var option = {
  ...,
  series: [{
    type: 'custom',
    renderItem: function (params, api) {
      var categoryIndex = api.value(0);
      var start = api.coord([api.value(1), categoryIndex]);
      var end = api.coord([api.value(2), categoryIndex]);
      var height = api.size([0, 1])[1] * 0.6;

      var rectShape = echarts.graphic.clipRectByRect({
        x: start[0],
        y: start[1] - height / 2,
        width: end[0] - start[0],
        height: height
      }, {
        x: params.coordSys.x,
        y: params.coordSys.y,
        width: params.coordSys.width,
        height: params.coordSys.height
      });

      return rectShape && {
        type: 'rect',
        shape: rectShape,
        style: api.style()
      };
    },
    data: data
  ]
}

```

renderItem wird für jedes Datenelement aufgerufen.

renderItem stellt zwei Parameter zur Verfügung:

- params: Gibt Informationen über momentane Serie, Daten und Koordinatensystem.
- api: Enthält einige Methoden.

renderItem sollte die Definition eines grafischen Elements zurückgeben. Siehe renderItem.return

Im Allgemeinen ist der Hauptprozess von renderItem Werte aus data zu holen und sie in grafische Elemente des momentanen Koordinatensystems umzuwandeln. Zwei Methoden in renderItem.arguments.api werden dabei immer benutzt:

- api.value(...) wird verwendet um Werte aus data zu holen. Zum Beispiel gibt api.value(0) den Wert der ersten Dimension des momentanen Datenelements zurück.
- api.coord(...) wird verwendet um Daten in Koordinaten umzuwandeln. Zum Beispiel wandelt var point = api.coord([api.value(0), api.value(1)]) die Daten zu einem Punkt im momentanen Koordinatensystem um.

Manchmal wird api.size(...) benötigt, was die Größe des Koordinatensystems nach einer gegebenen Datenspanne berechnet.

Außerdem kann api.style(...) verwendet werden um Stile zu setzen. Die Methode stellt nicht nur die Einstellungen zur Verfügung, die in series.itemStyle spezifiziert sind, sondern auch das Ergebnis des visuellen Mappings. Diese

Methode kann auch so aufgerufen werden: `api.style({fill: 'green', stroke: 'yellow'})` um diese Stileinstellungen zu überschreiben.

### 6.19.1286 `chart.series.i[].renderLabelForZeroData`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Beschriftungen für leere Daten gerendert werden müssen.

### 6.19.1287 `chart.series.i[].right`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der rechten Seite des Containers

Der Wert von `right` kann ein direkter Pixelwert wie `20` sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie `'20%'` sein.

Erlaubte Werte

- {integer}
- {percent}

### 6.19.1288 `chart.series.i[].roam`

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob das Zoomen und Bewegen mit der Maus aktiviert werden soll, standardmäßig false. Wenn entweder zoomen oder bewegen gewünscht ist, kann der Wert auf 'scale' oder 'move' gesetzt werden. Andernfalls kann er auf true gesetzt werden, um beides zu aktivieren.

#### Erlaubte Werte

- {boolean}
- scale
- move

### 6.19.1289 chart.series.i[].roseType

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Diagramm als Nightingale-Diagramm dargestellt werden soll, welches Daten durch den Radius unterscheidet. Es gibt zwei optionale Werte:

- 'radius': Benutzt den zentralen Winkel um den Prozentanteil der Daten zu zeigen, und den Radius um die Datengröße zu zeigen.
- 'area': Alle Sektoren haben den gleichen zentralen Winkel, Datengröße wird durch den Radius angezeigt.

#### Erlaubte Werte

- {boolean}
- radius
- area

### 6.19.1290 chart.series.i[].sampling

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Downsampling-Strategie, die verwendet wird, wenn die Datengröße deutlich höher als die Pixelgröße ist. Erhöht wenn eingeschaltet die Performanz. Standardmäßig abgeschaltet, wodurch alle Datenpunkte gezeichnet werden.

Mögliche Werte:

- 'average': Nutze durchschnittlichen Wert der Filterpunkte.
- 'max': Nutze maximalen Wert der Filterpunkte.
- 'min': Nutze minimalen Wert der Filterpunkte.
- 'sum': Nutze die Summe der Filterpunkte.

Erlaubte Werte

- average
- max
- min
- sum

### 6.19.1291 chart.series.i[].selectedMode

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

selectedMode bestimmt, ob multiple Auswahl unterstützt wird. Standardmäßig wird false verwendet, um die Auswahl zu deaktivieren.

Der Wert kann auch 'single', für die Auswahl einzelner Bereiche, oder 'multiple', für die Auswahl mehrerer Bereiche sein.

Erlaubte Werte

- {boolean}

- single
- multiple

### 6.19.1292 chart.series.i[].selectedOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Offset-Distanz des ausgewählten Vektors.

### 6.19.1293 chart.series.i[].seriesLayoutBy

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'column' ]

Wenn ein Datenset benutzt wird, bestimmt `seriesLayoutBy`, ob die Spalten oder Zeilen des Datensets auf die Serien gemappt werden. Also, ob die Serien ein Zeilen- oder Spaltenlayout haben. Mögliche Werte:

- 'column': Standardmäßig werden die Spalten des Datensets auf die Serien gemappt. In diesem Fall steht jede Spalte für eine Dimension.
- 'row': Die Zeilen des Datensets werden auf die Serien gemappt. In diesem Fall steht jede Zeile für eine Dimension.

#### Erlaubte Werte

- column
- row

### 6.19.1294 chart.series.i[].showAllSymbol

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Funktioniert nur, wenn die Hauptachse eine Kategorieachse ist (axis.type ist 'category').

Mögliche Werte:

- 'auto': Standardwert. Zeige alle Symbole, wenn genug Platz ist. Folge sonst der Intervallstrategie mit axisLabel.interval.
- true: Zeige alle Symbole.
- false: Folge der Intervallstrategie mit axisLabel.interval.

Erlaubte Werte

- {boolean}
- auto

### 6.19.1295 chart.series.i[].showEffectOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'render' ]

Spezifiziert, wann der Effekt angezeigt werden soll.

**Mögliche Werte:**

- 'render': Zeige den Effekt, wenn das Rendern abgeschlossen ist.
- 'emphasis': Zeige den Effekt, wenn das Element (durch die Maus) hervorgehoben wird.



## Erlaubte Werte

- render
- emphasis

## 6.19.1296 chart.series.i[].showSymbol

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Symbol angezeigt werden soll. Es würde während dem Zeigen auf den Tooltip angezeigt werden.

## 6.19.1297 chart.series.i[].silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Standardmäßig false, um Maus-Events auszulösen und auf sie zu reagieren.

## 6.19.1298 chart.series.i[].silent.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Aktiviere einen Hyperlink-Sprung, wenn auf den Knoten geklickt wird. Dieser Wert ist verfügbar, wenn `series-treemap.nodeClick` auf 'link' steht.

Siehe `series-treemap.data.target`.

### 6.19.1299 `chart.series.i[].silent.object`

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Maus-Events ignoriert werden sollen. Der Standardwert ist false, um Maus-Events auszulösen und auf solche zu reagieren.

### 6.19.1300 `chart.series.i[].silent.target`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Die gleiche Bedeutung wie `target` im HTML-Label `<a>`, siehe `series-treemap.data.link`. Mögliche Werte sind 'blank' und 'self'.

## Erlaubte Werte

- blank
- self

## 6.19.1301 chart.series.i[].silent.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

## 6.19.1302 chart.series.i[].silent.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenfarbe des schwebenden Tooltip-Layers.

## 6.19.1303 chart.series.i[].silent.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 6.19.1304 chart.series.i[].silent.tooltip.extraCSSText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 6.19.1305 chart.series.i[].silent.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

## 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

## 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 6.19.1306 chart.series.i[].silent.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen der Daten dieser Serie.

### 6.19.1307 chart.series.i[].silent.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.19.1308 chart.series.i[.silent.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```



**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

### 6.19.1309 chart.series.i[.].silent.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.19.1310 chart.series.i[].silent.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1311 chart.series.i[].silent.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.19.1312 chart.series.i[.silent.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.1313 chart.series.i[.silent.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.1314 chart.series.i[.].silent.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1315 chart.series.i[.].silent.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.19.1316 chart.series.i[].silent.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.data.tooltip funktioniert nur wenn tooltip.trigger auf 'item' steht.

Der Textil des schwebenden Tooltip-Layers.

### 6.19.1317 chart.series.i[].silent.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.1318 chart.series.i[].silent.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1319 chart.series.i[.].silent.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1320 chart.series.i[.].silent.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.19.1321 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 6.19.1322 chart.series.i[].silent.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.1323 chart.series.i[].silent.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1324 chart.series.i[].singleAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Index der einzelnen Zeitachse. Standardmäßig 0, da er nur eine Achse enthält.

### 6.19.1325 chart.series.i[].smooth

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob Kurven geglättet werden sollen.

Wird ein Boolean-Wert angegeben, so bestimmt er, ob Glättung angewendet wird. Wird ein Dezimalwert von 0 bis 1 angegeben, steht er für den Glättungsgrad. Kleinere Werte bedeuten geringere Glättung.

Siehe smoothMonotone um den Glättungsalgorithmus zu verändern.



### 6.19.1326 chart.series.i[].smoothMonotone

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die gebrochene Linie die Monotonität bei der Glättung behält. Der Wert kann auf 'x' oder 'y' gesetzt werden, um die Monotonität auf der X- oder Y-Achse zu behalten. Alternativ kann der Wert auf 'none' gesetzt werden, um einen nicht-monotonen Glättungsalgorithmus zu verwenden.

Der Standardmäßige Glättungsalgorithmus wurde von ECharts 4.0.3 verbessert. Der alte Algorithmus kann verwendet werden, indem smoothMonotone auf 'none' gesetzt wird.

Der alte Algorithmus verwendet den vorherigen und den nächsten Punkt um die Richtung des Kontrollpunkts zu bestimmen, während die Richtung beim neuen Algorithmus immer entweder horizontal (wenn die erste Dimension der Daten monoton ist) oder vertikal (wenn die zweite Dimension der Daten monoton ist) ist.

Der neue Algorithmus funktioniert jedoch nicht mit nicht-monotonen Daten.

Es wird also empfohlen in den meisten Situationen den Standardwert von smoothMonotone zu verwenden. Sind die Daten auf der Y-Achse monoton, sollte der Wert auf 'y' gesetzt werden. Sind die Daten nicht-monoton, sollte der Wert auf 'none' gesetzt werden, um den alten Algorithmus zu verwenden.

#### Erlaubte Werte

- x
- y
- none

### 6.19.1327 chart.series.i[].sort

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Sortiermethode.

**Sunburst:**

[ Default-Wert: 'desc' ]

Sortiermethode die von den Sektoren basierend auf value verwendet wird, was die Summe der Kinder ist, wenn nicht gesetzt. Der Standardwert 'desc' steht für absteigende Reihenfolge. Auch möglich ist 'asc' für aufsteigende Reihenfolge, null für keine Sortierung, oder eine Callback-Funktion wie:

```
function(nodeA, nodeB) {
  return nodeA.getValue() - nodeB.getValue();
}
```

**Funnel:**

[ Default-Wert: 'descending' ]

Datensortierung, akzeptiert die Werte 'ascending', 'descending', 'none' (keine Sortierung), oder eine Funktion nach `Array.prototype.sort(function(a, b) { ... });`

## 6.19.1328 chart.series.i[].splitLine.length

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 30 ]

Die Länge der Trennlinie. Kann ein Prozentwert relativ zum Radius sein.

## Erlaubte Werte

- {integer}
- {percent}

## 6.19.1329 chart.series.i[].splitLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #eee ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {  
  type: 'linear',  
  x: 0,  
  y: 0,  
  x2: 0,  
  y2: 1,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {  
  type: 'radial',  
  x: 0.5,  
  y: 0.5,  
  r: 0.5,  
  colorStops: [{  
    offset: 0, color: 'red' // Farbe bei 0% Position  
  }, {  
    offset: 1, color: 'blue' // Farbe bei 100% position  
  }],  
  globalCoord: false // Standardmäßig false  
}
```

```
// Fill with texture
```

```
color: {  
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.  
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.  
}
```

### 6.19.1330 chart.series.i[].splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Linienstil der Trennlinie.

### 6.19.1331 chart.series.i[].splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.19.1332 chart.series.i[].splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.19.1333 chart.series.i[].splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.19.1334 chart.series.i[].splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.19.1335 chart.series.i[].splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.19.1336 chart.series.i[].splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.19.1337 chart.series.i[].splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Breite der Trennlinie.

## 6.19.1338 chart.series.i[].splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Trennlinie

## 6.19.1339 chart.series.i[].splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinie angezeigt werden soll.

## 6.19.1340 chart.series.i[].splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Die Anzahl der getrennten Segmente der Skala des Messdiagramms.

### 6.19.1341 chart.series.i[].squareRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der erwartete quadratische Schnitt. Das Layout versucht, sich so nah wie möglich an diesen Schnitt anzupassen. Standardmäßig verwendet wird der goldenene Schnitt:  $0.5 * (1 + \text{Math.sqrt}(5))$ .

### 6.19.1342 chart.series.i[].stack

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Name des Stacks. Auf der gleichen Kategorieachse werden Serien mit dem gleichen Stack-Namen aufeinandergelegt.

### 6.19.1343 chart.series.i[].startAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Anfangswinkel, dessen Wert zwischen 0 und 360 liegt.

Die Stelle direkt rechts vom Mittelpunkt ist 0 Grad, über dem Mittelpunkt 90 Grad, links vom Mittelpunkt 180 Grad.

### 6.19.1344 chart.series.i[].step

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Linie als Schrittlinie dargestellt werden soll. Der Wert 'start', 'middle' oder 'end' sein, was den Schrittpunkt der Schrittlinie bestimmt. Alternativ kann der Wert true oder false sein, wobei true identisch zu 'start' ist.

#### Erlaubte Werte

- {boolean}
- start
- middle
- end

### 6.19.1345 chart.series.i[].stillShowZeroSum

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob Sektoren angezeigt werden sollen, wenn alle Daten 0 sind.

### 6.19.1346 chart.series.i[].symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbol der Serie.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

### 6.19.1347 chart.series.i[].symbolBoundingData

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert eine Grenzfläche, die für die grafischen Elemente zur Verfügung steht. Diese Einstellung gibt einen Datenpunkt, welcher dann in eine Koordinate im Koordinatensystem übersetzt wird. Die Koordinate spezifiziert die Grenze. Insbesondere hängt, wenn `symbolBoundingData` gesetzt wurde, die finale Größe (oder das Layout) des grafischen Elements von `symbolBoundingData` ab.

Ist die Referenzleiste horizontal, korrespondiert `symbolBoundingData` zur X-Achse. Ist die Referenzleiste vertikal, korrespondiert `symbolBoundingData` zur Y-Achse.

Regel:

- Wenn `symbolRepeat` nicht verwendet wird:  
`symbolBoundingData` ist standardmäßig so groß wie die Referenzleiste. Die Größe des grafischen Elements wird durch `symbolBoundingData` bestimmt. Zum Beispiel, wenn die Referenzleiste vertikal und ihr Wert 24 ist, `symbolSize` auf `[30, '50%']` und `symbolBoundingData` auf 124 gesetzt ist, wird die endgültige Größe des grafischen Elements  $124 * 50\% = 62$  sein. Ist `symbolBoundingData` nicht gesetzt, wird die endgültige Größe  $24 * 50\% = 12$  sein.
- Wenn `symbolRepeat` verwendet wird:  
`symbolBoundingData` ist der Extremwert des Koordinatensystems. `symbolBoundingData` definiert einen begrenzten Bereich, in dem wiederholte grafische Elemente anhand von `symbolMargin`, `symbolRepeat` und `symbolSize` ausgerichtet werden. Beide dieser Einstellungen bestimmen die Größe der Lücken der wiederholten grafischen Elemente.

`symbolBoundingData` wird normalerweise in den folgenden Fällen verwendet:

- Wenn `symbolClip` verwendet wird:  
Und eine Serie wird verwendet um "Anzahl Wert" anzuzeigen, während eine andere Serie verwendet wird, um "momentaner Wert" anzuzeigen. `symbolBoundingData` kann verwendet werden, um sicherzustellen, dass die grafischen Elemente dieser beiden Serien die gleiche Größe haben.
- Wenn `symbolRepeat` verwendet wird:  
`symbolBoundingData` kann verwendet werden, um sicherzustellen, dass die Lücken zwischen den Elementen in verschiedenen Säulen gleich groß sind. Natürlich kann `symbolBoundingData` auch nicht gesetzt werden, da der Standardwert ein stabiler Wert ist (Extremwert des Koordinatensystems).

`symbolBoundingData` kann auch ein Array sein, wie `[-40, 60]`, welches negative und positive Werte für `symbolBoundingData` spezifiziert.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolBoundingData: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolBoundingData: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolBoundingData: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

## 6.19.1348 chart.series.i[].symbolClip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob grafische Elemente zurechtgeschnitten werden sollen.

- false/null/undefined: Das gesamte grafische Element repräsentiert die Größe von value.
- true: Das zurechtgeschnittene grafische Element repräsentiert die Größe von value.

symbolClip wird normalerweise in diesem Fall benutzt: Sowohl ein maximaler Wert als auch ein momentaner Wert sollen angezeigt werden. In diesem Fall können zwei Serien verwendet werden. Eine für den maximalen Wert im Hintergrund, welche komplette Grafiken verwendet, eine für den momentanen Wert im Vordergrund, welche zurechtgeschnittene Grafiken verwendet.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolClip: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolClip: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolClip: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

## 6.19.1349 chart.series.i[].symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 6.19.1350 chart.series.i[].symbolMargin

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den Rand auf beiden Seiten eines grafischen Elements. ("Beide Seiten" bezeichnet die Seiten in Richtung der Wertachse.) Funktioniert nur, wenn symbolRepeat verwendet wird.

Es können absolute Werte (wie 20) verwendet werden, oder Prozentwerte (wie '-30%') relativ zu symbolSize.

symbolMargin kann ein positiver oder negativer Wert sein, was das Überlappen von grafischen Elementen ermöglicht, wenn symbolRepeat verwendet wird.

Ein "!" kann an den Wert angehängt werden, wie '30%!' oder '25!', wodurch eine zusätzliche Lücke an beiden Enden hinzugefügt wird. Sonst erreichen die grafischen Elemente auf beiden Seiten standardmäßig die Grenze.

Hinweis:

- Wenn symbolRepeat auf true/'fixed' gesetzt ist: Der gegebene symbolMargin ist nur ein Referenzwert. Der finale Rand der grafischen Elemente wird anhand von symbolRepeat, symbolMargin und symbolBoundingData berechnet.
- Wenn symbolRepeat auf eine Zahl gesetzt ist: symbolMargin funktioniert nicht mehr.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```

series: [{
  symbolMargin: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolMargin: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolMargin: ... // Betrifft nur dieses Datenelement.
  }]
}]

```

### 6.19.1351 chart.series.i[].symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

#### Erlaubte Werte

- {integer}
- {percent}

### 6.19.1352 chart.series.i[].symbolPatternSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 400 ]

Ein Bild kann als Muster von grafischen Elementen verwendet werden.

```
var textureImg = new Image();
textureImg.src = 'data:image/jpeg;base64,...'; // dataURI
// Oder
// textureImg.src = 'http://xxx.xxx.xxx/xx.png'; // URL
...
itemStyle: {
  color: {
    image: textureImg,
    repeat: 'repeat'
  }
}
```

symbolPatternSize spezifiziert die Größe des Bildmusters. Ist symbolPatternSize zum Beispiel 400, wird das Bildmuster mit einer Größe von 400px \* 400px dargestellt.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolPatternSize: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolPatternSize: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolPatternSize: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### 6.19.1353 chart.series.i[].symbolRepeat

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob grafische Elemente wiederholt werden sollen. Mögliche Werte:

- false/null/unidentified: Keine Wiederholung. Das heißt, jedes grafische Element repräsentiert je ein Datenelement.
- true: Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen wird anhand der Daten berechnet.
- Eine Zahl: Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen ist immer die gegebene Zahl.
- 'fixed': Wiederholung. Das heißt, eine Gruppe an wiederholten grafischen Elementen repräsentiert ein Datenelement. Die Anzahl der Wiederholungen wird anhand von `symbolBoundingData` berechnet. Das heißt, die Anzahl der Wiederholungen hat nichts mit den Daten zu tun. Diese Option ist nützlich, wenn grafische Elemente als Hintergrund benutzt werden sollen.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolRepeat: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolRepeat: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolRepeat: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### Erlaubte Werte

- {boolean}
- {integer}
- fixed



## 6.19.1354 chart.series.i[].symbolRepeatDirection

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Wenn `symbolRepeat` verwendet wird, spezifiziert `symbolRepeatDirection` die Reihenfolge, in der die wiederholten grafischen Elemente gerendert werden. Diese Einstellung ist in den folgenden Fällen nützlich:

- Wenn `symbolMargin` auf einen negativen Wert gesetzt ist, überlappen sich wiederholte Elemente. `symbolRepeatDirection` kann verwendet werden, um die Reihenfolge der Überlappungen zu bestimmen.
- Wird `animationDelay` oder `animationDelayUpdate` verwendet, bestimmt `symbolRepeatDirection` die Reihenfolge der Indices.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von `series-pictorialBar.data` gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolRepeatDirection: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolRepeatDirection: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolRepeatDirection: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

### Erlaubte Werte

- start
- end

### 6.19.1355 chart.series.i[].symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 6.19.1356 chart.series.i[].symbolSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolgröße.

Mögliche Werte:

#### **effectScatter:**

Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

#### **line, scatter, radar, tree, graph:**

Der Wert kann als einzelne Zahl, wie 10, gesetzt werden, oder ein Array kann verwendet werden, um Breite und Höhe darzustellen. Zum Beispiel steht [20, 10] für eine Symbolbreite von 20 und eine Symbolhöhe von 10.

Müssen die Größen der Symbole unterschiedlich sein, kann eine Callback-Funktion im folgenden Format verwendet werden:

```
(value: Array|number, params: Object) => number|Array
```

Der erste Parameter value ist der Wert in data, der zweite Parameter params enthält die restlichen Parameter vom data-Objekt.

**lines:**

Symbolgröße an den beiden Enden der Linie. Der Wert kann ein Array sein, um beide Enden separat zu definieren, oder ein String, um beide Enden zusammen zu definieren.

Vorsicht: Es können nicht Breite und Höhe getrennt gesetzt werden wie beim normalen SymbolSize.

**pictorialBar:**

Der Wert kann als Array der Form [Breite, Höhe] gesetzt werden. Zum Beispiel steht [20, 10] für eine Breite von 20 und eine Höhe von 10. Er kann auch als einzelne Zahl gesetzt werden, zum Beispiel 10, was äquivalent zu [10, 10] ist.

Absolute Werte (wie 10) und prozentuale Werte (wie '120%', ['55%', 23]) können verwendet werden.

Werden prozentuale Werte verwendet, so wird die endgültige Größe des grafischen Elements anhand der Referenzleiste ausgerechnet.

Als Beispiel gebe es eine Referenzleiste, die auf der X-Achse basiert (selber also eine vertikale Leiste ist) und sei symbolSize auf ['30%', '50%'] gesetzt. Die endgültige Größe des grafischen Elements ist:

- Breite: <Breite der Referenzleiste> \* 30%.
- Höhe:
  - Wenn symbolRepeat verwendet wird: <Höhe der Referenzleiste> \* 50%.
  - Wenn symbolRepeat nicht verwendet wird: <Breite der Referenzleiste> \* 50%.

Der Fall für die Y-Achse verläuft analog.

Dieses Attribut kann auf der Wurzelebene einer Serie definiert werden, wo alle Datenelemente dieser Serie davon betroffen sind. Dieses Attribut kann auch für jedes Element von series-pictorialBar.data gesetzt werden, wo nur dieses Element davon betroffen ist.

Zum Beispiel:

```
series: [{
  symbolSize: ... // Betrifft alle Datenelemente.
  data: [23, 56]
}]
// Or
series: [{
  data: [{
    value: 23
    symbolSize: ... // Betrifft nur dieses Datenelement.
  }, {
    value: 56
    symbolSize: ... // Betrifft nur dieses Datenelement.
  }]
}]
```

## 6.19.1357 chart.series.i[].title.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.19.1358 chart.series.i[].title.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.19.1359 chart.series.i[].title.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.1360 chart.series.i[].title.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1361 chart.series.i[].title.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Textfarbe.

### 6.19.1362 chart.series.i[].title.fontFamily

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1363 chart.series.i[].title.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Schriftgröße.

### 6.19.1364 chart.series.i[].title.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

## Erlaubte Werte

- normal
- italic
- oblique

## 6.19.1365 chart.series.i[].title.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.1366 chart.series.i[].title.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.19.1367 chart.series.i[].title.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.19.1368 chart.series.i[].title.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Der Titel des Messdiagramms.

### 6.19.1369 chart.series.i[].title.offsetCenter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, '-40%'] ]

Die Offset-Position relativ zum Mittelpunkt des Messdiagramms. Das erste Element des Arrays ist der horizontale Offset, das zweite Element ist der vertikale Offset. Kann in absoluten Werten oder Prozentwerten relativ zum Radius des Messdiagramms angegeben werden.

Erlaubte Werte

- {integer}
- {percent}

### 6.19.1370 chart.series.i[].title.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.19.1371 chart.series.i[].title.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `n` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

## 6.19.1372 chart.series.i[].title.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.1373 chart.series.i[].title.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.1374 chart.series.i[].title.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1375 chart.series.i[].title.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1376 chart.series.i[].title.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Titel angezeigt werden soll.

### 6.19.1377 chart.series.i[].title.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.1378 chart.series.i[].title.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1379 chart.series.i[].title.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1380 chart.series.i[].title.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1381 chart.series.i[].title.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1382 chart.series.i[].title.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1383 chart.series.i[].title.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1384 chart.series.i[].tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 6.19.1385 chart.series.i[].tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 6.19.1386 chart.series.i[].tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 6.19.1387 chart.series.i[].tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.



```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 6.19.1388 chart.series.i[].tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

#### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

#### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

#### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 6.19.1389 chart.series.i[].tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen dieser Serie.

### 6.19.1390 chart.series.i[].tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.19.1391 chart.series.i[.].tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 6.19.1392 chart.series.i[].tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.19.1393 chart.series.i[].tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1394 chart.series.i[].tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.19.1395 chart.series.i[].tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.1396 chart.series.i[].tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal

- bold
- bolder
- lighter
- {integer}

### 6.19.1397 chart.series.i[].tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1398 chart.series.i[].tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:



```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.19.1399 chart.series.i[].tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

**Hinweis:** series.tooltip funktioniert nur, wenn tooltip.trigger auf 'item' steht.

Der Textstil des schwebenden Tooltip-Layers.

### 6.19.1400 chart.series.i[].tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.19.1401 chart.series.i[].tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
-----------------	---------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1402 chart.series.i[].tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1403 chart.series.i[].tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.19.1404 chart.series.i[].tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 6.19.1405 chart.series.i[].tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

## 6.19.1406 chart.series.i[].tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1407 chart.series.i[].top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Abstand zwischen der Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.19.1408 chart.series.i[].upperLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.19.1409 chart.series.i[].upperLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1410 chart.series.i[].upperLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1411 chart.series.i[].upperLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.19.1412 chart.series.i[].upperLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.19.1413 chart.series.i[].upperLabel.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1414 chart.series.i[].upperLabel.distance

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Abstand zum übergeordneten grafischen Element. Funktioniert wenn position einen String-Wert hat (wie 'top', 'insideRight').

### 6.19.1415 chart.series.i[].upperLabel.ellipsis

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob überlaufender Text durch Punkte ersetzt werden soll.

### 6.19.1416 chart.series.i[].upperLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.19.1417 chart.series.i[].upperLabel.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------



<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.19.1418 chart.series.i[.upperLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.19.1419 chart.series.i[.upperLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.19.1420 chart.series.i[.upperLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Formatierer der Datenbeschriftung, unterstützt String-Templates und Callback-Funktionen. In beiden Formen wird `\n` als Repräsentation einer neuen Zeile unterstützt.

### String-Templates

Die Modellvariablen umfassen:

- {a}: Serienname.
- {b}: Name des Datenelements.
- {c}: Wert des Datenelements.
- {@xxx}: Der Wert einer Dimension namens 'xxx', so steht {@product} für den Wert der Dimension 'product'.
- {@[n]}: Der Wert der Dimension bei Index n, so steht {@[3]} für den Wert von dimensions[3].

### Beispiel:

```
formatter: '{b}: {@score}'
```

### Callback-Funktionen

Callback-Funktionen liegen vor in der Form

```
(params: Object|Array) => string
```

wobei params das Datenset ist, das vom Formatierer benötigt wird, und die folgende Form hat:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Datenwert
  value: number|Array,
  // Datenfarbe
  color: string,
}
```

### 6.19.1421 chart.series.i[].upperLabel.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 20 ]

Höhe des Bereichs der Beschriftung.

### 6.19.1422 chart.series.i[].upperLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.19.1423 chart.series.i[].upperLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

upperLabel wird verwendet, um zu spezifizieren, ob eine Beschriftung angezeigt werden soll, wenn der Knoten Kinder hat. Wenn upperLabel.show auf true gesetzt ist, wird die Beschriftung von Elternknoten angezeigt.

So wie series-treemap.label kann die Option upperLabel direkt an der Wurzel von series-treemap definiert werden, in series-treemap.level, oder in jedem Element von series-treemap.data.

Anders gesagt spezifiziert series-treemap.label den Stil der Beschriftung von Blattknoten, während series-treemap.upperLabel den Stil der Beschriftung von Knoten mit Kindern spezifiziert. Die Beschriftung wird dann im oberen inneren Bereich des Knotens angezeigt.

Tips: In treemap kann das upperLabel Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.1424 chart.series.i[].upperLabel.offset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob der Text leicht verschoben werden soll. Zum Beispiel: [30, 40] steht für eine Bewegung um 30 horizontal und um 40 vertikal.

### 6.19.1425 chart.series.i[].upperLabel.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.19.1426 chart.series.i[].upperLabel.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung

#### Kann die folgenden Werte annehmen:

- [x, y]  
Benutze relative Prozentwerte oder absolute Pixelwerte um die Position der Beschriftung relativ zur oberen linken Ecke der Bounding-Box zu beschreiben. Zum Beispiel:

```
// Absolute Pixelwerte
position: [10, 10],
// Relative Prozentwerte
position: ['50%', '50%']
```

- String, der die Position der Beschriftung relativ zur Bounding-Box beschreibt.

#### Erlaubte Werte

- top
- left
- right
- bottom
- inside
- insideLeft
- insideRight
- insideTop
- insideBottom
- insideTopLeft
- insideBottomLeft
- insideTopRight
- insideBottomRight
- {integer}
- {percent}

### 6.19.1427 chart.series.i[.].upperLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.19.1428 chart.series.i[.upperLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

## 6.19.1429 chart.series.i[].upperLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.19.1430 chart.series.i[].upperLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.19.1431 chart.series.i[].upperLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.19.1432 chart.series.i[].upperLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.19.1433 chart.series.i[].upperLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.19.1434 chart.series.i[].upperLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.19.1435 chart.series.i[].upperLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.19.1436 chart.series.i[].upperLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.19.1437 chart.series.i[].upperLabel.textShadowColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.19.1438 chart.series.i[.].upperLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.19.1439 chart.series.i[.].upperLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.19.1440 chart.series.i[].upperLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.19.1441 chart.series.i[].upperLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.19.1442 chart.series.i[].visibleMin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Ein Knoten wird nicht angezeigt wenn die Größe seines Bereichs kleiner als dieser Wert ist (Einheit: px quadrat).

So werden kleine Knoten, die ansonsten eng aneinander stehen würden, versteckt. Wenn ein Nutzer die treemap vergrößert, vergrößert sich auch die Bereichsgröße, und das Rechteck wird angezeigt, wenn sie über diese Grenze steigt.

Siehe serie-treemap.levels für Details zu visuellem Encoding.

Tips: In treemap kann das visibleMin Attribut an mehreren Stellen vorkommen:

- In series-treemap, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von series-treemap.levels, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von series-treemap.data, wo es die Einstellungen der einzelnen Knoten bestimmt.

### 6.19.1443 chart.series.i[].visualDimension

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

treemap kann beliebige Dimensionen der Daten ins Visuelle mappen.

Der Wert von `series-treemap.data` kann ein Array sein. Jeder Eintrag des Arrays steht für eine "dimension". `visualDimension` spezifiziert die Dimension, auf der visuelles Mapping durchgeführt wird.

Siehe `series-treemap.levels` für Details zu visuellem Encoding.

Tips: In `treemap` kann das `visualDimension` Attribut an mehreren Stellen vorkommen:

- In `series-treemap`, wo es die vereinte Einstellung der Serie bestimmt.
- In jedem Array-Element von `series-treemap.levels`, wo es die vereinte Einstellung der Elemente dieser Ebene bestimmt.
- In jedem Knoten von `series-treemap.data`, wo es die Einstellungen der einzelnen Knoten bestimmt.

#### 6.19.1444 `chart.series.i[].visualMax`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn `colorMappingBy` auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von `visualMin` oder `visualMax` bestimmt werden.

#### 6.19.1445 `chart.series.i[].visualMin`

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: null ]

Der minimale Wert der momentanen Ebene. Standardmäßig automatisch.

Wenn colorMappingBy auf 'value' steht, kann das Ausmaß des visuellen Mappings manuell durch setzen von visualMin oder visualMax bestimmt werden.

## 6.19.1446 chart.series.i[].width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite der Komponente.

## 6.19.1447 chart.series.i[].xAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der X-Achse mit der kombiniert werden soll, nützlich für mehrere X-Achsen in einem Diagramm.

## 6.19.1448 chart.series.i[].yAxisIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Index der X-Achse mit der kombiniert werden soll, nützlich für mehrere X-Achsen in einem Diagramm.

### 6.19.1449 chart.series.i[].z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.19.1450 chart.series.i[].zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.



zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.19.1451 chart.series.i[.].zoom

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Zoom-Rate des momentanen Viewports.

### 6.19.1452 chart.series.i[.].zoomToNodeRatio

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.32\*0.32 ]

Die treemap wird automatisch auf eine angemessene Rate gezoomt, wenn ein Knoten angeklickt wird (solange nodeClick auf 'zoomToNode' gesetzt ist und kein Drill-Down durchgeführt wird). Diese Einstellung gibt die Rate an.

### 6.19.1453 chart.series.i[.].silent.children

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Series

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kindknoten, rekursive definition. Konfigurationen sind identisch zu denen in series.data

### 6.19.1454 chart.series.fixed.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sunburst' ]

Wird in Sunburst-Diagrammen verwendet, um einen Namen für die Serie bereitzustellen.

Example:

```
dashlet.addOption("chart.series.fixed.name", "Gesamt");
```

### 6.19.1455 chart.series.i[].name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	gantt, series-name

Name für eine Series

Beispiel:

```
// For Gantt chart, give name for the different series, to be used in click-function
(for example) dashlet.addOption("chart.series.n[ganttitem].name", "cmganttdiagramm");
dashlet.addOption("chart.series.n[ganttyaxis1].name", "GantBars");
dashlet.addOption("chart.series.n[ganttyaxis2].name", "pdficon");
```

### 6.19.1456 chart.series.i[].label.alignTo

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Default: none

Label-Ausrichtungsrichtlinie.

- 'none' (default): Etikettenzeilen haben eine feste Länge als `labelLine.length` und `labelLine.length2`
- 'labelLine': Die Ausrichtung am Ende der Labelzeilen und die Länge der kürzesten horizontalen Labelzeilen wird durch `labelLine.length2` konfiguriert.
- 'edge': Die Ausrichtung auf Text und der Abstand zwischen den Rändern des Textes und dem Ansichtsfenster wird durch `label.edgeDistance` konfiguriert.

#### Erlaubte Werte

- labelLine
- edge
- none

### 6.19.1457 chart.series.i[].label.edgeDistance

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Der horizontale Abstand zwischen Textkanten und Ansichtsfenster, wenn label.position „outer“ und label.alignTo „edge“ ist.

In den meisten Fällen benötigen Sie einen kleinen edgeDistance-Wert wie 10 für mobile Geräte, um sicherzustellen, dass mehr Text angezeigt werden kann, anstatt .... Bei größeren Auflösungen sollte jedoch ein Prozentwert angewendet werden, damit die Beschriftungszeilen nicht angezeigt werden zu lang. Wenn Ihr Diagramm in unterschiedlichen Auflösungen verwendet wird, ist es ratsam, responsives Design für unterschiedliche Auflösungen festzulegen.

### 6.19.1458 chart.series.i[].label.bleedMargin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Der horizontale Abstand zwischen Text und Ansichtsfenster, wenn label.position „outer“ und label.alignTo „none“ oder „labelLine“ ist. Längerer Text wird in '...' abgeschnitten.

### 6.19.1459 chart.series.i[].label.distanceToLabelLine

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Abstand zwischen Beschriftungslinie und Text.

### 6.19.1460 chart.series.i[].minShowLabelAngle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Series
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.21
<b>Schlagworte / Tags</b>	

Wenn ein Sektor kleiner als dieser Winkel ist (0 ~ 360), werden Label und LabelLine nicht angezeigt.

## 6.20 ECharts (5) Single Axis

### 6.20.1 chart.singleAxis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Eine Achse mit einer einzelnen Dimension. Kann verwendet werden um Daten in einer Dimension darzustellen.

### 6.20.2 chart.singleAxis.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.20.3 chart.singleAxis.zlevel

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

#### 6.20.4 chart.singleAxis.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

#### 6.20.5 chart.singleAxis.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- right
- center

### 6.20.6 chart.singleAxis.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- top
- middle

- bottom

## 6.20.7 chart.singleAxis.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Erlaubte Werte

- {percent}
- {integer}

## 6.20.8 chart.singleAxis.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '5%' ]

Abstand zwischen der Grid-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Erlaubte Werte

- {percent}



- {integer}

### 6.20.9 chart.singleAxis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Grid-Komponente. Standardmäßig adaptiv.

### 6.20.10 chart.singleAxis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Grid-Komponente. Standardmäßig adaptiv.

### 6.20.11 chart.singleAxis.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'horizontal' ]

Orientierung der Achse. Standardmäßig ist sie horizontal, man kann diesen Wert auf 'vertical' setzen um eine vertikale Achse zu erhalten.

#### Erlaubte Werte

- horizontal
- vertical

## 6.20.12 chart.singleAxis.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'value' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.
- 'log': Logarithmische Achse, geeignet für logarithmische Daten.

#### Erlaubte Werte

- value
- category
- time
- log

## 6.20.13 chart.singleAxis.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name der Achse.

### 6.20.14 chart.singleAxis.nameLocation

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'start' ]

Position des Achsennamens.

Erlaubte Werte

- start
- middle
- center
- end

### 6.20.15 chart.singleAxis.nameTextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil des Achsennamens.

### 6.20.16 chart.singleAxis.nameTextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe des Achsennamens. Verwendet standardmäßig axisLine.lineStyle.color.

### 6.20.17 chart.singleAxis.nameTextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil des Achsennamens.

Erlaubte Werte

- normal
- italic
- oblique

### 6.20.18 chart.singleAxis.nameTextStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke des Achsennamens.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.20.19 chart.singleAxis.nameTextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie des Achsennamens.

Akzeptiert auch 'serif' , 'monospace' , ...

### 6.20.20 chart.singleAxis.nameTextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße des Achsennamens.

### 6.20.21 chart.singleAxis.nameTextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.20.22 chart.singleAxis.nameTextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.20.23 chart.singleAxis.nameTextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.20.24 chart.singleAxis.nameTextStyle.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.20.25 chart.singleAxis.nameTextStyle.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.20.26 chart.singleAxis.nameTextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.20.27 chart.singleAxis.nameTextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.20.28 chart.singleAxis.nameTextStyle.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.20.29 chart.singleAxis.nameTextStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.20.30 chart.singleAxis.nameTextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.20.31 chart.singleAxis.nameTextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.20.32 chart.singleAxis.nameTextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.20.33 chart.singleAxis.nameTextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.34 chart.singleAxis.nameTextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.35 chart.singleAxis.nameTextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.20.36 chart.singleAxis.nameTextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.20.37 chart.singleAxis.nameTextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.20.38 chart.singleAxis.nameTextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.20.39 chart.singleAxis.nameTextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.20.40 chart.singleAxis.nameTextStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.20.41 chart.singleAxis.nameTextStyle.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

#### 6.20.42 chart.singleAxis.nameGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Freiraum zwischen Achsenname und Achsenlinie.

### 6.20.43 chart.singleAxis.nameRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rotation des Achsennamens.

### 6.20.44 chart.singleAxis.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achse invertiert werden soll. Neue Option aus ECharts 3.

### 6.20.45 chart.singleAxis.boundaryGap

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Grenzabstand auf beiden Seiten einer Koordinatenachse. Der Datentyp und das Verhalten unterscheiden sich für Kategorieachsen und Nicht-Kategorieachsen.

Der Grenzabstand einer Kategorieachse kann auf entweder true oder false gesetzt werden. Der Standardwert ist true, wodurch axisTick als reine Trennlinie dient, und Daten sowie Label nur in den Zwischenräumen zwischen Ticks, auch Bänder genannt, erscheinen.

Für Nicht-Kategorieachsen, wie numerische, Zeit- und Protokollachsen, ist boundaryGap ein Array aus zwei Werten, welches die Spannweite zwischen dem minimalen und maximalen Wert darstellt. Die Werte sind entweder numerische Werte oder relative Prozentwerte, welche ungültig werden, falls min und max gesetzt sind. **Beispiel:**

```
boundaryGap: ['20%', '20%']
```

#### 6.20.46 chart.singleAxis.min

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der minimale Wert der Achse.

Kann auf einen speziellen Wert 'dataMin' gesetzt werden, wodurch der minimale Wert der Achse auf das minimale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.20.47 chart.singleAxis.max

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der maximale Wert der Achse.

Kann auf einen speziellen Wert 'dataMax' gesetzt werden, wodurch der maximale Wert der Achse auf das maximale Label gesetzt wird.

Falls der Wert nicht gesetzt wird, wird er automatisch so gewählt, dass die Achsenticks gleichmäßig verteilt sind.

Bei einer Kategorieachse kann der Wert auch auf eine Ordnungszahl gesetzt werden. Zum Beispiel, hat eine Kategorieachse die Daten ['kategorieA', 'kategorieB', 'kategorieC'], so repräsentiert die Ordnungszahl 2 'kategorieC'. Außerdem kann er auf negative Zahlen gesetzt werden, wie -3.

### 6.20.48 chart.singleAxis.scale

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der Wert ist nur für numerische Achsen, also Typ 'value', verfügbar.

Er spezifiziert, ob die Nullposition der Achse obligatorisch dargestellt werden soll. Wird der Wert auf true gesetzt, so kann die Achse die Nullposition weglassen, was in Streudiagrammen auf beiden Wertachsen nützlich ist.

Diese Option ist nicht verfügbar wenn min und max gesetzt sind.

### 6.20.49 chart.singleAxis.splitNumber

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Anzahl der Segmente in die die Achse aufgespalten wird. Es ist anzumerken, dass diese Zahl nur als Empfehlung dient, und die tatsächliche Anzahl Segmente aus Gründen der Lesbarkeit angepasst werden kann.

Diese Option ist für Kategorieachsen nicht verfügbar.

### 6.20.50 chart.singleAxis.minInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Minimaler Abstand zwischen Trennlinien.

Zum Beispiel kann der Wert bei einer Zeitachse (Typ ist 'time') auf  $3600 * 24 * 1000$  gesetzt werden, um sicherzustellen, dass der Abstand zwischen den Trennlinien größer gleich einem Tag ist.

```
{
  minInterval: 3600 * 1000 * 24
}
```

Die Option ist nur für Achsen vom Typ 'value' oder 'time' verfügbar.

### 6.20.51 chart.singleAxis.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Obligatorischer Segmentationsintervall für die Achse.

Da splitNumber nur ein Empfehlungswert ist, liegen die berechneten Ticks potentiell nicht wie erwartet. In so einem Fall sollte interval zusammen mit min und max verwendet werden, um obligatorische Ticks zu setzen. In den meisten Fällen ist die Verwendung dieser Option aber nicht empfohlen, da die automatische Berechnung für die meisten Fälle ausreicht.

Diese Option ist für Kategorieachsen nicht verfügbar. Für Achsen mit Typ 'time' sollte ein Zeitstempel übergeben werden. Für Achsen mit Typ 'log' sollte ein logarithmierter Wert übergeben werden.

### 6.20.52 chart.singleAxis.logBase

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Basis des Logarithmus, was nur für numerische Achsen mit Typ 'log' gültig ist.

### 6.20.53 chart.singleAxis.silent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

True für Achsen mit denen nicht interagiert werden kann.

## 6.20.54 chart.singleAxis.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftungen der Achse Maus-Events auslösen und auf diese reagieren.

Die Parameter für Events enthalten:

```
{
  // Komponententyp: xAxis, yAxis, radiusAxis, angleAxis
  // Wovon jeder ein Attribut für den Index hat, z.B. xAxisIndex für xAxis
  componentType: string,
  // Wert der Achse vor der Formatierung.
  // Wertlabel anklicken um Event auszulösen.
  value: '',
  // Name der Achse.
  // Namenlabel anklicken um das Event auszulösen.
  name: ''
}
```

## 6.20.55 chart.singleAxis.axisLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu axisLine gehören.

### 6.20.56 chart.singleAxis.axisLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie gezeigt werden soll.

### 6.20.57 chart.singleAxis.axisLine.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'none' ]

Symbole der beiden Enden der Achse. Der Wert kann ein String sein, welcher das gleiche Symbol für beide Enden repräsentiert, oder ein Array aus zwei Strings, welches die Symbole der Enden separat repräsentiert. Der Wert ist standardmäßig 'none', wodurch an beiden Enden kein Pfeil steht. Steht der Wert auf 'arrow', so steht an beiden Enden ein Pfeil. Soll nur am hinteren Ende ein Pfeil stehen, so sollte der Wert ['none', 'arrow'] sein.

#### Erlaubte Werte

- none
- arrow

### 6.20.58 chart.singleAxis.axisLine.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [10, 15] ]

Größe der Pfeile an den beiden Enden. Der erste Wert ist die Breite im rechten Winkel zur Achse, der nächste die Breite parallel zur Achse.

### 6.20.59 chart.singleAxis.axisLine.symbolOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Abstand der Pfeile zur Achse. Ist der Wert ein Array, so ist der erste Wert der Abstand des Pfeils am Anfang und der zweite Wert der Abstand des Pfeils am Ende. Ist der Wert eine Zahl, haben beide Pfeile diesen gleichen Abstand.

### 6.20.60 chart.singleAxis.axisLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenlinie.

## 6.20.61 chart.singleAxis.axisLine.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #333]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.



```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 6.20.62 chart.singleAxis.axisLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.20.63 chart.singleAxis.axisLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.64 chart.singleAxis.axisLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.65 chart.singleAxis.axisLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.66 chart.singleAxis.axisLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.67 chart.singleAxis.axisLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.68 chart.singleAxis.axisLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.69 chart.singleAxis.axisTick.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zu den Achsenticks gehören.

### 6.20.70 chart.singleAxis.axisTick.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenticks angezeigt werden sollen.

### 6.20.71 chart.singleAxis.axisTick.alignWithLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Richte die Achsenlabel anhand der Achsenticks aus, was nur verfügbar ist wenn boundaryGap aus der Kategorie axis auf true gesetzt ist.

### 6.20.72 chart.singleAxis.axisTick.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.20.73 chart.singleAxis.axisTick.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenticks nach innen gerichtet werden sollen. Standardmäßig false.

### 6.20.74 chart.singleAxis.axisTick.length

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Länge des Achsenticks.

### 6.20.75 chart.singleAxis.axisTick.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achsenticks.

### 6.20.76 chart.singleAxis.axisTick.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenticks. Standardmäßig auf axisLine.lineStyle.color gesetzt.

### 6.20.77 chart.singleAxis.axisTick.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Achsenticks.

### 6.20.78 chart.singleAxis.axisTick.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Achsenticks.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.79 chart.singleAxis.axisTick.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.80 chart.singleAxis.axisTick.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.



### 6.20.81 chart.singleAxis.axisTick.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.82 chart.singleAxis.axisTick.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.83 chart.singleAxis.axisTick.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.84 chart.singleAxis.axisLabel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellungen, die zur Achsenbeschriftung gehören.

### 6.20.85 chart.singleAxis.axisLabel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenbeschriftungen angezeigt werden sollen.

### 6.20.86 chart.singleAxis.axisLabel.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

### 6.20.87 chart.singleAxis.axisLabel.inside

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Achsenbeschriftungen im Inneren des Diagramms platziert werden sollen. Standardmäßig false.

### 6.20.88 chart.singleAxis.axisLabel.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rotationsgrad der Achsenbeschriftungen, was besonders nützlich ist wenn für eine Kategorieachse nicht genug Platz vorhanden ist.

Der Rotationsgrad geht von -90 bis 90.

### 6.20.89 chart.singleAxis.axisLabel.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 8 ]

Der Abstand zwischen den Achsenbeschriftungen und der Achsenlinie.

### 6.20.90 chart.singleAxis.axisLabel.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Benutzung von String-Templates; Die Template-Variable ist die standardmäßige
Achenbeschriftung {value}
formatter: '{value} kg'

// Benutzung von Callback-Funktionen; Die Funktionsparameter sind standardmäßige
Achenbeschriftung und index
formatter: function (value, index) {
  // Formatiert auf Monat/Tag; Zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 6.20.91 chart.singleAxis.axisLabel.showMinLabel

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des kleinsten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des kleinsten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.20.92 chart.singleAxis.axisLabel.showMaxLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Spezifiziert, ob die Beschriftung des größten Ticks angezeigt werden soll. Mögliche Werte: true, false, null. Standardmäßig wird automatisch entschieden, wodurch die Beschriftung des größten Ticks nicht angezeigt wird, falls die Beschriftungen sich überlappen.

### 6.20.93 chart.singleAxis.axisLabel.color

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Farbe der Achsenbeschriftung, ist standardmäßig `axisLine.lineStyle.color`. Callback-Funktionen werden im folgenden Format unterstützt:

```
(val: string) => Color
```

Der Parameter ist der Text der Beschriftung und der Rückgabewert ist die Farbe. Siehe das folgende Beispiel:

```
textStyle: {
  color: function (value, index) {
    return value >= 0 ? 'green' : 'red';
  }
}
```

### 6.20.94 chart.singleAxis.axisLabel.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.20.95 chart.singleAxis.axisLabel.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

## 6.20.96 chart.singleAxis.axisLabel.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

## 6.20.97 chart.singleAxis.axisLabel.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

## 6.20.98 chart.singleAxis.axisLabel.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.20.99 chart.singleAxis.axisLabel.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

#### Erlaubte Werte

- top
- middle
- bottom

### 6.20.100 chart.singleAxis.axisLabel.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.20.101 chart.singleAxis.axisLabel.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```
backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}
```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.20.102 chart.singleAxis.axisLabel.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.20.103 chart.singleAxis.axisLabel.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.20.104 chart.singleAxis.axisLabel.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.20.105 chart.singleAxis.axisLabel.padding

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.20.106 chart.singleAxis.axisLabel.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.20.107 chart.singleAxis.axisLabel.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.20.108 chart.singleAxis.axisLabel.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.20.109 chart.singleAxis.axisLabel.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.20.110 chart.singleAxis.axisLabel.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.111 chart.singleAxis.axisLabel.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.112 chart.singleAxis.axisLabel.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.20.113 chart.singleAxis.axisLabel.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.20.114 chart.singleAxis.axisLabel.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.20.115 chart.singleAxis.axisLabel.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.20.116 chart.singleAxis.axisLabel.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.20.117 chart.singleAxis.axisLabel.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.20.118 chart.singleAxis.axisLabel.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```
label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}
```

### 6.20.119 chart.singleAxis.splitLine.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennlinie der Achse im Gitterbereich.

### 6.20.120 chart.singleAxis.splitLine.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Trennlinien angezeigt werden sollen. Auf numerischen Achsen werden sie standardmäßig angezeigt, auf Kategorieachsen nicht.

### 6.20.121 chart.singleAxis.splitLine.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.20.122 chart.singleAxis.splitLine.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennlinien.

### 6.20.123 chart.singleAxis.splitLine.lineStyle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#ccc']]

Die Farbe der Trennlinie, welche separat gesetzt werden kann.

Die Farbe kann auch in einem Farbarray gesetzt werden, dessen Werte nacheinander von den Trennlinien verwendet werden.

Beispiel:

```
splitLine: {
  lineStyle: {
    // Dunkle und helle Farben werden abwechselnd verwendet
    color: ['#aaa', '#ddd']
  }
}
```

### 6.20.124 chart.singleAxis.splitLine.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Breite der Trennlinien.

### 6.20.125 chart.singleAxis.splitLine.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Typ der Trennlinie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.126 chart.singleAxis.splitLine.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.127 chart.singleAxis.splitLine.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.128 chart.singleAxis.splitLine.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.129 chart.singleAxis.splitLine.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.130 chart.singleAxis.splitLine.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.131 chart.singleAxis.splitArea.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Trennbereiche der Achse im Gitterbereich, standardmäßig nicht angezeigt.

### 6.20.132 chart.singleAxis.splitArea.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

### 6.20.133 chart.singleAxis.splitArea.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Trennbereiche angezeigt werden sollen.

### 6.20.134 chart.singleAxis.splitArea.areaStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Trennbereiche.

### 6.20.135 chart.singleAxis.splitArea.areaStyle.color

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['rgba(250,250,250,0.3)', 'rgba(200,200,200,0.3)'] ]

Farbe der Trennbereiche. Die Farbe kann auch in einem Farbarray definiert werden, dessen Einträge nacheinander von den Trennbereichen verwendet werden. Standardmäßig werden abwechselnd dunkle und helle Farben verwendet.

### 6.20.136 chart.singleAxis.splitArea.areaStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.137 chart.singleAxis.splitArea.areaStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.138 chart.singleAxis.splitArea.areaStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.139 chart.singleAxis.splitArea.areaStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.140 chart.singleAxis.splitArea.areaStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.141 chart.singleAxis.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Einstellung die zum axisPointer der Achse gehören.

### 6.20.142 chart.singleAxis.axisPointer.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Der axisPointer wird standardmäßig nicht angezeigt. Wurde tooltip.trigger jedoch auf 'axis' oder tooltip.axisPointer.type auf 'cross' gesetzt, so wird der axisPointer automatisch angezeigt. Jedes Koordinatensystem wählt automatisch die Achsen aus, die ihre axisPointer anzeigen. tooltip.axisPointer.axis kann verwendet werden, um die Wahl zu verändern.

### 6.20.143 chart.singleAxis.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatortyp.

Optionen:

- 'line' Linienindikator.
- 'shadow' Schattierter Fadenkreuz-Indikator.
- 'none' Kein Indikator angezeigt.

Erlaubte Werte

- line
- shadow
- none

### 6.20.144 chart.singleAxis.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.20.145 chart.singleAxis.axisPointer.z

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.20.146 chart.singleAxis.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung von axisPointer.

### 6.20.147 chart.singleAxis.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Beschriftung angezeigt werden soll. Standardmäßig wird die Beschriftung nicht angezeigt. Ist aber tooltip.axisPointer.type auf 'cross' gesetzt, so wird die Beschriftung automatisch angezeigt.

### 6.20.148 chart.singleAxis.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes in der Beschriftung. Wird standardmäßig automatisch bestimmt. Der Wert kann auch auf '2' gesetzt werden, was anzeigt, dass zwei Stellen hinter dem Komma reserviert werden.

### 6.20.149 chart.singleAxis.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

**Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.20.150 chart.singleAxis.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 6.20.151 chart.singleAxis.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#fff" ]

Textfarbe.

### 6.20.152 chart.singleAxis.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 6.20.153 chart.singleAxis.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.20.154 chart.singleAxis.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]



Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.20.155 chart.singleAxis.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.20.156 chart.singleAxis.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.20.157 chart.singleAxis.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.158 chart.singleAxis.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.159 chart.singleAxis.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.20.160 chart.singleAxis.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.20.161 chart.singleAxis.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.20.162 chart.singleAxis.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.20.163 chart.singleAxis.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.20.164 chart.singleAxis.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.20.165 chart.singleAxis.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.20.166 chart.singleAxis.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.20.167 chart.singleAxis.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.20.168 chart.singleAxis.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.20.169 chart.singleAxis.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.170 chart.singleAxis.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.171 chart.singleAxis.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.172 chart.singleAxis.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.173 chart.singleAxis.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type auf 'line' steht.

### 6.20.174 chart.singleAxis.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.20.175 chart.singleAxis.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.20.176 chart.singleAxis.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.177 chart.singleAxis.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.178 chart.singleAxis.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.179 chart.singleAxis.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.180 chart.singleAxis.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.181 chart.singleAxis.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.182 chart.singleAxis.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type auf 'shadow' steht.

### 6.20.183 chart.singleAxis.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.20.184 chart.singleAxis.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.185 chart.singleAxis.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.186 chart.singleAxis.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.187 chart.singleAxis.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.188 chart.singleAxis.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.189 chart.singleAxis.axisPointer.triggerTooltip

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]



Spezifiziert, ob der Tooltip ausgelöst werden soll.

### 6.20.190 chart.singleAxis.axisPointer.value

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Momentaner Wert. Wenn axisPointer.handle verwendet wird, kann value gesetzt werden, um die initiale Position von axisPointer zu bestimmen.

### 6.20.191 chart.singleAxis.axisPointer.status

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Momentaner Status, kann 'show' oder 'hide' sein

Erlaubte Werte

- show
- hide

### 6.20.192 chart.singleAxis.axisPointer.handle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ein Knopf mit dem der axisPointer gezogen werden kann. Dieses Feature findet in Geräten mit Berührungssteuerung Verwendung.

### 6.20.193 chart.singleAxis.axisPointer.handle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Auf true setzen, um den Handle zu verwenden.

### 6.20.194 chart.singleAxis.axisPointer.handle.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon des Handles.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.20.195 chart.singleAxis.axisPointer.handle.size

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 45 ]

Die Größe des Handles, welche als einzelner Wert oder als Array ([Breite, Höhe]) gesetzt werden kann.

### 6.20.196 chart.singleAxis.axisPointer.handle.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Abstand von der Mitte des Handles zur Achse.

### 6.20.197 chart.singleAxis.axisPointer.handle.color

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Farbe des Handles.

### 6.20.198 chart.singleAxis.axisPointer.handle.throttle

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 40 ]

Drosselt die Rate der Aktualisierungen der Ansicht wenn der Handle bewegt wird, in ms. Man kann diesen Wert erhöhen um die Performanz zu verbessern und die Nutzererfahrung zu verschlechtern.

### 6.20.199 chart.singleAxis.axisPointer.handle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.200 chart.singleAxis.axisPointer.handle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.201 chart.singleAxis.axisPointer.handle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.202 chart.singleAxis.axisPointer.handle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.203 chart.singleAxis.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip-Einstellungen in der Komponente des Koordinatensystems.

#### Grundlegende Einführung:

Der Tooltip kann an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

### 6.20.204 chart.singleAxis.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

### 6.20.205 chart.singleAxis.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentyten in grid und polar unterstützt. Weiterhin lassen sich Achsen über axisPointer.axis zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item
- axis
- none

### 6.20.206 chart.singleAxis.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Konfigurationselement für den Achsenindikator.

`tooltip.axisPointer` ist wie syntaktischer Zucker von `axisPointer`-Einstellungen an Achsen (zum Beispiel `xAxis.axisPointer` oder `angleAxis.axisPointer`). Detailliertere Einstellungen können an `someAxis.axisPointer` vorgenommen werden. Im Normalfall ist die Verwendung von `tooltip.axisPointer` jedoch praktischer.

**Hinweis:** Konfigurationen an `tooltip.axisPointer` haben eine niedrigere Priorität als `someAxis.axisPointer`

## 6.20.207 `chart.singleAxis.tooltip.axisPointer.type`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatorotyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei `axisPointers` auf zwei orthometrischen Achsen zu aktivieren.

Erlaubte Werte

- line
- shadow
- none
- cross

## 6.20.208 `chart.singleAxis.tooltip.axisPointer.axis`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem selbst die Achsen, welche ihren Achsenpointer anzeigen sollen (Standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

#### Erlaubte Werte

- x
- y
- radius
- angle

### 6.20.209 chart.singleAxis.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

### 6.20.210 chart.singleAxis.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.20.211 chart.singleAxis.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Beschriftung des axisPointers.

### 6.20.212 chart.singleAxis.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.20.213 chart.singleAxis.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

## 6.20.214 chart.singleAxis.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

#### Parameter:

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

#### 6.20.215 chart.singleAxis.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 6.20.216 chart.singleAxis.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.20.217 chart.singleAxis.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal

- italic
- oblique

### 6.20.218 chart.singleAxis.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.20.219 chart.singleAxis.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.20.220 chart.singleAxis.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.20.221 chart.singleAxis.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.20.222 chart.singleAxis.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.223 chart.singleAxis.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.224 chart.singleAxis.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.20.225 chart.singleAxis.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.20.226 chart.singleAxis.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

## 6.20.227 chart.singleAxis.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

## 6.20.228 chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

## 6.20.229 chart.singleAxis.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.20.230 chart.singleAxis.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [5, 7, 5, 7] ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Die Positionen können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.20.231 chart.singleAxis.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.20.232 chart.singleAxis.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.20.233 chart.singleAxis.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.20.234 chart.singleAxis.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.235 chart.singleAxis.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.236 chart.singleAxis.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.237 chart.singleAxis.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.238 chart.singleAxis.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig, wenn axisPointer.type auf 'line' steht.

### 6.20.239 chart.singleAxis.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.20.240 chart.singleAxis.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite.

### 6.20.241 chart.singleAxis.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Stil der Linie.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.242 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.243 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.244 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.245 chart.singleAxis.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.246 chart.singleAxis.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.247 chart.singleAxis.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

## 6.20.248 chart.singleAxis.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.20.249 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.250 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.251 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.252 chart.singleAxis.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.253 chart.singleAxis.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.254 chart.singleAxis.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

## 6.20.255 chart.singleAxis.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.20.256 chart.singleAxis.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.20.257 chart.singleAxis.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.20.258 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.20.259 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.20.260 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.20.261 chart.singleAxis.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.20.262 chart.singleAxis.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.20.263 chart.singleAxis.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLDomElement, rect:
Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'

Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

- 'bottom'

Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 6.20.264 chart.singleAxis.tooltip.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

### 6.20.265 chart.singleAxis.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

### 6.20.266 chart.singleAxis.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

### 6.20.267 chart.singleAxis.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

### 6.20.268 chart.singleAxis.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```



## 6.20.269 chart.singleAxis.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

## 6.20.270 chart.singleAxis.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

## 6.20.271 chart.singleAxis.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.20.272 chart.singleAxis.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

## 6.20.273 chart.singleAxis.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.20.274 chart.singleAxis.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.20.275 chart.singleAxis.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.20.276 chart.singleAxis.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.277 chart.singleAxis.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.20.278 chart.singleAxis.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.20.279 chart.singleAxis.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.20.280 chart.singleAxis.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.20.281 chart.singleAxis.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.20.282 chart.singleAxis.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.20.283 chart.singleAxis.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.20.284 chart.singleAxis.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Single Axis
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

## 6.21 ECharts (5) Text Style

### 6.21.1 chart.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Globaler Schriftstil.

### 6.21.2 chart.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.21.3 chart.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.21.4 chart.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
-----------------	---------



<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter

### 6.21.5 chart.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.21.6 chart.textStyle.fontSize

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.21.7 chart.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.21.8 chart.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.21.9 chart.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.21.10 chart.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.21.11 chart.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.21.12 chart.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.21.13 chart.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.21.14 chart.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.21.15 chart.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.21.16 chart.textStyle.overflow

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

Erlaubte Werte

- truncate
- break
- breakAll

### 6.21.17 chart.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Text Style
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

## 6.22 ECharts (5) Timeline

### 6.22.1 chart.timeline.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeistrahle-Komponente, welche Funktionen bereitstellt wie das Wechseln und das Abspielen von ECharts Optionen.

Anders als andere Komponenten braucht die Zeitstrahl-Komponente mehrere Optionen. Wenn die traditionelle Art der ECharts Option als *atomic option* bezeichnet wird, dann sollte die für Zeitstrahlen verwendete Option *compound option* genannt werden, da sie aus mehreren zusammengestellten atomic options besteht. Zum Beispiel:

```
// Im folgenden Beispiel ist baseOption eine "atomic option", genau wie alle Einträge
// im options Array.
// Jede der atomic options folgt einer Konfiguration, die in diesem Dokument
// eingeführt wird.
myChart.setOption(
  {
    baseOption: {
      timeline: {
        ...,
        data: ['2002-01-01', '2003-01-01', '2004-01-01']
      },
      title: {
        subtext: ' Daten sind vom National Bureau of Statistics '
      },
      grid: {...},
      xAxis: [...],
      yAxis: [...],
      series: [
        { // Andere Einstellungen von Serie 1
          type: 'bar',
          ...
        },
        { // Andere Einstellungen von Serie 2
          type: 'line',
          ...
        },
        { // Andere Einstellungen von Serie 3
          type: 'pie',
          ...
        }
      ]
    },
    options: [
      { // Option, die zu '2002-01-01' gehört
        title: {
          text: 'Die Statistiken aus dem Jahr 2002'
        },
        series: [
          {data: []}, // Die Daten von Serie 1
          {data: []}, // Die Daten von Serie 2
          {data: []} // Die Daten von Serie 3
        ]
      },
      { // Option, die zu '2003-01-01' gehört
        title: {
          text: 'Die Statistiken aus dem Jahr 2003'
        },
        series: [
          {data: []},
          {data: []},
          {data: []}
        ]
      }
    ]
  }
)
```



```

    },
    { // Option, die zu '2004-01-01' gehört
      title: {
        text: 'Die Statistiken aus dem Jahr 2004'
      },
      series: [
        {data: []},
        {data: []},
        {data: []}
      ]
    }
  ]
}
);

```

Im obigen Beispiel gehört jeder Eintrag in `timeline.data` zu einer Option in Array von `options`.

### Hinweise und Best Practices

- Geteilte Einstellungen sollten in `baseOption` gesetzt werden. Wenn der Punkt auf dem Zeitstrahl gewechselt wird, wird die zugehörige Option im `options` Array mit `baseOption` kombiniert, um die Endgültige Option zu erhalten.
- Ist ein Attribut in einer Option im `option` Array konfiguriert, sollte es auch bei den anderen Optionen konfiguriert sein. Ist es das nicht, wird es ignoriert.
- Optionen in *compound options* unterstützen keine Vereinigung. Das bedeutet, dass beim erneuten Aufruf von `chart.setOption(rawOption)`, falls `rawOption` eine *compound option* ist (also ein `options` Array enthält), die neuen `rawOption.options` die alten ersetzen, statt sich mit ihnen zu vereinen. Das neue `rawOption.baseOption` wird hingegen mit dem Alten vereint.

### Kompatibilität mit ECharts 2:

- ECharts 3 unterstützt den Parameter `timeline.notMerge` nicht mehr, und somit implizit auch den *notMerge-Modus* nicht. Wird diese Funktion gebraucht, kann die Option im eigenen Programm bearbeitet werden, bevor sie an `setOption(option, true)` übergeben wird.
- Im Vergleich zwischen ECharts 3 und ECharts 2 ist der Definitionsort der Zeitstrahl-Attribute ein Anderer. In ECharts 3 wurde er zu `baseOption` umgelegt und wird als separate Komponente angesehen, welche auch mit dem Definitionsort von ECharts 2 kompatibel ist, was aber nicht empfohlen wird.

## 6.22.2 chart.timeline.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Zeitstrahl-Komponente angezeigt werden soll. Steht der Wert auf false, wird sie nicht angezeigt, führt aber trotzdem ihre Funktionen aus.

### 6.22.3 chart.timeline.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'slider' ]

Dieses Attribut hat momentan 'slider' als einzigen gültigen Wert. Muss nicht verändert werden.

### 6.22.4 chart.timeline.axisType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'time' ]

Achsentyp.

Mögliche Werte:

- 'value': Numerische Achse, geeignet für kontinuierliche Daten.
- 'category': Kategorieachse, geeignet für diskrete kategorische Daten. Daten sollten für diesen Typ nur über data gesetzt werden.
- 'time': Zeitachse, geeignet für kontinuierliche Daten von Zeitserien. Im Vergleich zur numerischen Achse hat diese hier eine bessere Formatierung für Zeit und eine andere Methode zur Tickberechnung. Zum Beispiel entscheidet sie sich für die Ticks abhängig von der Zeitspanne für Monate, Wochen, Tage oder Stunden.

Erlaubte Werte

- value
- category

- time

### 6.22.5 chart.timeline.currentIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Zeigt an, welches Element momentan ausgewählt ist. Ist currentIndex zum Beispiel 0, so ist der momentan ausgewählte Eintrag timeline.data[0] (unter Verwendung von options[0]).

### 6.22.6 chart.timeline.autoPlay

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die automatische Abspielung aktiviert ist.

### 6.22.7 chart.timeline.rewind

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob rückwärts Abspulen unterstützt wird.

### 6.22.8 chart.timeline.loop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob das Abspielen sich wiederholen soll.

### 6.22.9 chart.timeline.playInterval

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2000 ]

Gibt die Abspielgeschwindigkeit (Zeit zwischen den Wechseln) in Millisekunden an.

### 6.22.10 chart.timeline.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert wird, während der Kontrollpunkt sich bewegt.

### 6.22.11 chart.timeline.controlPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'left' ]

Position des "Abspielen"-Knopfes

Erlaubte Werte

- left
- right

### 6.22.12 chart.timeline.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.22.13 chart.timeline.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.22.14 chart.timeline.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- left
- center
- right
- {percent}
- {integer}

### 6.22.15 chart.timeline.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

### Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom

### 6.22.16 chart.timeline.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 6.22.17 chart.timeline.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Zeitstrahl-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

Erlaubte Werte

- {percent}
- {integer}

### 6.22.18 chart.timeline.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

## 6.22.19 chart.timeline.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Ausrichtung der Komponente, gültige Werte sind:

- 'vertical': Vertikales Layout.
- 'horizontal': Horizontales Layout.

### Erlaubte Werte

- vertical
- horizontal

## 6.22.20 chart.timeline.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Zeitstrahl-Komponente umgekehrt werden soll, wodurch das erste Element zum letzten wird usw.

## 6.22.21 chart.timeline.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'emptyCircle' ]

Symbol des Zeitstrahls.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Erlaubte Werte

- circle
- emptyCircle
- rect
- roundRect
- triangle

- diamond
- pin
- arrow
- none

### 6.22.22 chart.timeline.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe des Zeitstrahl-Symbols. Der Wert kann als einzelne Zahl gesetzt werden, wie 10, oder als Array welches für Breite und Höhe steht. Zum Beispiel: [20, 10] steht für ein Symbol mit Breite 20 und Höhe 10.

### 6.22.23 chart.timeline.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationgrad des Zeitstrahl-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 6.22.24 chart.timeline.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 6.22.25 chart.timeline.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 6.22.26 chart.timeline.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil der Achse des Zeitstrahls.

### 6.22.27 chart.timeline.lineStyle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Achsenlinie angezeigt werden soll. Kann auf false gesetzt werden um die Achsenlinie zu verstecken um einen anderen Stil zu verwenden.

### 6.22.28 chart.timeline.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Linienfarbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.22.29 chart.timeline.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 2 ]

Linienbreite des Zeitstrahls.

## 6.22.30 chart.timeline.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp des Zeitstrahls.

### Erlaubte Werte

- solid
- dashed
- dotted

## 6.22.31 chart.timeline.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.22.32 chart.timeline.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.22.33 chart.timeline.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.22.34 chart.timeline.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.22.35 chart.timeline.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.22.36 chart.timeline.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Achsenbeschriftung, emphasis ist der Stil des hervorgehobenen Textes. Der Textstil in emphasis wird zum Beispiel verwendet, wenn die Maus auf den Text zeigt oder die Legende eine Verbindung schließt.

### 6.22.37 chart.timeline.label.position

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Mögliche Werte:

- 'auto': Automatisches Layout.
- 'left': An den linken Rand stellen. Gültig wenn timeline.orient auf 'horizontal' steht.
- 'right': An den rechten Rand stellen. Gültig wenn timeline.orient auf 'horizontal' steht.
- 'top': An den oberen Rand stellen. Gültig wenn timeline.orient auf 'vertical' steht.
- 'bottom': An den unteren Rand stellen. Gültig wenn timeline.orient auf 'vertical' steht.
- Nummer: Wird ein numerischer Wert zugewiesen, steht dieser für den Freiraum zwischen Beschriftung und Achse. Ist der Wert 0, steht die Beschriftung an der gleichen Stelle wie die Achse. Negative Werte sind gültig, um Freiraum in die andere Richtung zu definieren.

Erlaubte Werte

- auto
- left
- right
- top
- bottom
- {integer}

### 6.22.38 chart.timeline.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.22.39 chart.timeline.label.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Intervall der Beschriftungen. Wird der Wert auf einen numerischen Wert wie 2 gesetzt, wird alle zwei Elemente eine Beschriftung angezeigt.

### 6.22.40 chart.timeline.label.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.22.41 chart.timeline.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achsenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Verwendung von String-Templates; Die Template-Variablen {value} ist die
// standardmäßige Beschriftung der Achse
formatter: '{value} kg'

// Verwendung von Callback-Funktionen; Funktionsparameter sind Achsenwert und
// Achsenindex
formatter: function (value, index) {
  // Formatiert auf Monat/Tag, zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getFullYear());
  }
  return texts.join('/');
}
```

## 6.22.42 chart.timeline.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#304654" ]

Textfarbe.

## 6.22.43 chart.timeline.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

#### 6.22.44 chart.timeline.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

#### 6.22.45 chart.timeline.label.fontFamily

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

timeline.lable.normal Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.22.46 chart.timeline.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.22.47 chart.timeline.label.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### Erlaubte Werte

- left
- center
- right

## 6.22.48 chart.timeline.label.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.22.49 chart.timeline.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.22.50 chart.timeline.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.  
 Oder ein Bild kann verwendet werden, zum Beispiel:



```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

### 6.22.51 chart.timeline.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

### 6.22.52 chart.timeline.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.22.53 chart.timeline.label.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.22.54 chart.timeline.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.22.55 chart.timeline.label.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.22.56 chart.timeline.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.22.57 chart.timeline.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.22.58 chart.timeline.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.22.59 chart.timeline.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.22.60 chart.timeline.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.22.61 chart.timeline.label.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

## 6.22.62 chart.timeline.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

## 6.22.63 chart.timeline.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.22.64 chart.timeline.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.22.65 chart.timeline.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.22.66 chart.timeline.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.22.67 chart.timeline.label.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.22.68 chart.timeline.label.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textuelle Beschriftung der hervorgehobenen Zeitstrahl-Komponente, um Informationen über die Daten von grafischen Elementen zu geben, wie Wert, Name und so weiter. label wurde in ECharts 2.x unter itemStyle platziert. In ECharts 3 wurde label auf eine Ebene mit itemStyle gebracht, um die Konfigurationsstruktur flacher zu machen, und hat wie itemStyle einen emphasis-Eintrag.



### 6.22.69 chart.timeline.label.emphasis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Beschriftung angezeigt werden soll.

### 6.22.70 chart.timeline.label.emphasis.interval

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Intervall der Beschriftungen. Wird der Wert auf einen numerischen Wert wie 2 gesetzt, wird alle zwei Elemente eine Beschriftung angezeigt.

### 6.22.71 chart.timeline.label.emphasis.rotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Beschriftung rotieren, von -90 Grad bis 90, positive Werte stehen für Rotation gegen den Uhrzeigersinn.

### 6.22.72 chart.timeline.label.emphasis.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Formatierer der Achsenbeschriftung, welcher String-Templates und Callback-Funktionen unterstützt.

Beispiel:

```
// Verwendung von String-Templates; Die Template-Variable {value} ist die
// standardmäßige Beschriftung der Achse
formatter: '{value} kg'

// Verwendung von Callback-Funktionen; Funktionsparameter sind Achsenwert und
// Achsenindex
formatter: function (value, index) {
  // Formatiert auf Monat/Tag, zeige Jahr nur in der ersten Beschriftung an
  var date = new Date(value);
  var texts = [(date.getMonth() + 1), date.getDate()];
  if (idx === 0) {
    texts.unshift(date.getYear());
  }
  return texts.join('/');
}
```

### 6.22.73 chart.timeline.label.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Textfarbe.

### 6.22.74 chart.timeline.label.emphasis.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

Erlaubte Werte

- normal
- italic
- oblique

### 6.22.75 chart.timeline.label.emphasis.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.22.76 chart.timeline.label.emphasis.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

timeline.lable.emphasis Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.22.77 chart.timeline.label.emphasis.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.22.78 chart.timeline.label.emphasis.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right

### 6.22.79 chart.timeline.label.emphasis.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Wenn `verticalAlign` im `rich` nicht gesetzt wird, wird das `verticalAlign` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### Erlaubte Werte

- top
- middle
- bottom

### 6.22.80 chart.timeline.label.emphasis.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn `lineHeight` im `rich` nicht gesetzt wird, wird die `lineHeight` der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.22.81 chart.timeline.label.emphasis.backgroundColor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Textfragments.

Kann ein Farbstring sein, wie '#123234', 'red', 'rgba(0,23,11,0.3)'.

Oder ein Bild kann verwendet werden, zum Beispiel:

```

backgroundColor: {
  image: 'xxx/xxx.png'
  // Kann die URL eines Bildes sein,
  // oder dataURI,
  // oder HTMLImageElement,
  // oder HTMLCanvasElement.
}

```

width und height können bei der Verwendung eines Bildes spezifiziert werden, oder werden standardmäßig automatisch angepasst.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

## 6.22.82 chart.timeline.label.emphasis.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textfragments.

Wird der Wert auf 'auto' gesetzt, so wird die Farbe als visuelle Farbe zugewiesen, zum Beispiel durch die Serienfarbe.

### 6.22.83 chart.timeline.label.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textfragments.

### 6.22.84 chart.timeline.label.emphasis.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenradius des Textfragments.

### 6.22.85 chart.timeline.label.emphasis.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

## 6.22.86 chart.timeline.label.emphasis.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

## 6.22.87 chart.timeline.label.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

## 6.22.88 chart.timeline.label.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.22.89 chart.timeline.label.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.22.90 chart.timeline.label.emphasis.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.22.91 chart.timeline.label.emphasis.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.22.92 chart.timeline.label.emphasis.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.22.93 chart.timeline.label.emphasis.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.22.94 chart.timeline.label.emphasis.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.22.95 chart.timeline.label.emphasis.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.22.96 chart.timeline.label.emphasis.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.22.97 chart.timeline.label.emphasis.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.22.98 chart.timeline.label.emphasis.rich.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

## 6.22.99 chart.timeline.itemStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Grafikstil des Zeitstrahls, emphasis ist der Stil des hervorgehobenen Elements, der zum Beispiel verwendet wird, wenn die Maus auf das Objekt zeigt oder die Legende eine Verbindung schließt.

## 6.22.100 chart.timeline.itemStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Farbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.22.101 chart.timeline.itemStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



### Schlagworte / Tags

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

## 6.22.102 chart.timeline.itemStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

## 6.22.103 chart.timeline.itemStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

### Erlaubte Werte

- dashed
- dotted
- solid

### 6.22.104 chart.timeline.itemStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.22.105 chart.timeline.itemStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.22.106 chart.timeline.itemStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.22.107 chart.timeline.itemStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.22.108 chart.timeline.itemStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.22.109 chart.timeline.itemStyle.emphasis.object

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil von Element des Zeitstrahls im hervorgehobenen Zustand.

### 6.22.110 chart.timeline.itemStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Zeitstrahls.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Mit Textur füllen
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.22.111 chart.timeline.itemStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.22.112 chart.timeline.itemStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht wird kein Rahmen angezeigt.

### 6.22.113 chart.timeline.itemStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.22.114 chart.timeline.itemStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.22.115 chart.timeline.itemStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.22.116 chart.timeline.itemStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.22.117 chart.timeline.itemStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.22.118 chart.timeline.itemStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.22.119 chart.timeline.checkpointStyle.object

<b>Datentyp</b>	OBJECT
-----------------	--------



<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des ausgewählten Zeitstrahl-Elements (Checkpoint).

### 6.22.120 chart.timeline.checkpointStyle.symbol

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'circle' ]

Symbol des Checkpoints.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### Erlaubte Werte

- circle
- emptyCircle
- rect
- roundRect
- triangle
- diamond
- pin
- arrow
- none

### 6.22.121 chart.timeline.checkpointStyle.symbolSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Größe des Checkpoint-Symbols. Der Wert kann als einzelne Zahl gesetzt werden, wie 10, oder als Array welches für Breite und Höhe steht. Zum Beispiel: [20, 10] steht für ein Symbol mit Breite 20 und Höhe 10.

### 6.22.122 chart.timeline.checkpointStyle.symbolRotate

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Rotationsgrad des Checkpoint-Symbols. Es ist zu beachten, dass, wenn das Symbol in markLine auf 'arrow' gesetzt wurde, symbolRotate ignoriert wird und immer der Tangentenwinkel verwendet wird.

### 6.22.123 chart.timeline.checkpointStyle.symbolKeepAspect

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Aspekt für Symbole in der Form von path:// behalten werden soll.

### 6.22.124 chart.timeline.checkpointStyle.symbolOffset

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [0, 0] ]

Offset des Zeitstrahl-Symbols zur originalen Position. Standardmäßig werden Symbole in die Zentrale Position der Daten gesetzt. Ist das Symbol aber aus einem nutzerdefinierten Vektor-Pfad oder Bild, kann dies nicht vorausgesetzt werden. In diesem Fall kann dieses Attribut verwendet werden um einen Offset zur Standard-Position hinzuzufügen. Der Wert kann in absoluten Pixelwerten oder in relativen Prozentwerten sein.

Zum Beispiel, [0, '50%'] steht für einen Offset um 50 Prozent der Höhe des Symbols nach unten.

### 6.22.125 chart.timeline.checkpointStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Checkpoints in der Zeitstrahl-Komponente.

### 6.22.126 chart.timeline.checkpointStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Die Rahmenbreite des Checkpoints in der Zeitstrahl-Komponente.

### 6.22.127 chart.timeline.checkpointStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(194, 53, 49, 0.5)' ]

Die Rahmenfarbe des Checkpoints in der Zeitstrahl-Komponente.

### 6.22.128 chart.timeline.checkpointStyle.animation

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob in der Zeitstrahl-Komponente die Bewegung des Checkpoints von Punkt zu Punkt animiert ist.

### 6.22.129 chart.timeline.checkpointStyle.animationDuration

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 300 ]

Die Animationsdauer des Checkpoints in der Zeitstrahl-Komponente.

### 6.22.130 chart.timeline.checkpointStyle.animationEasing

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'quinticInOut' ]

Der Linderungseffekt der Checkpoint-Animation in der Zeitstrahl-Komponente.

#### Erlaubte Werte

- linear
- quadraticIn
- quadraticOut
- quadraticInOut
- cubicIn
- cubicOut
- cubicInOut
- quarticIn
- quarticOut
- quarticInOut
- quinticIn
- quinticOut
- quinticInOut
- sinusoidalIn

- sinusoidalOut
- sinusoidalInOut
- exponentialIn
- exponentialOut
- exponentialInOut
- circularIn
- circularOut
- circularInOut
- elasticIn
- elasticOut
- elasticInOut
- backIn
- backOut
- backInOut
- bounceIn
- bounceOut
- bounceInOut

### 6.22.131 chart.timeline.controlStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Stil der Kontrollknöpfe, was beinhaltet: *Wiedergabe*, *Vorheriger* und *Nächster*.

### 6.22.132 chart.timeline.controlStyle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Kontrollknöpfe angezeigt werden sollen. False um alle zu verbergen.

### 6.22.133 chart.timeline.controlStyle.showPlayBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Wiedergabe-Knopf angezeigt werden soll.

### 6.22.134 chart.timeline.controlStyle.showPrevBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Vorheriger-Knopf angezeigt werden soll.

### 6.22.135 chart.timeline.controlStyle.showNextBtn

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der Nächster-Knopf angezeigt werden soll.

### 6.22.136 chart.timeline.controlStyle.itemSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 22 ]

Größe der Kontrollknöpfe, in Pixeln (px).

### 6.22.137 chart.timeline.controlStyle.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Intervall zwischen den Kontrollknöpfen, in Pixeln (px).

### 6.22.138 chart.timeline.controlStyle.position

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'left' ]

Die Platzierung der Kontrollknöpfe.

- Wenn `timeline.orient` auf 'horizontal' steht, sind 'left' und 'right' gültig.
- Wenn `timeline.orient` auf 'vertical' steht, sind 'top' und 'bottom' gültig.

Erlaubte Werte

- left
- right
- top
- bottom

### 6.22.139 `chart.timeline.controlStyle.playIcon`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Abspielen* Status des *Abspielen* Knopfs

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.22.140 `chart.timeline.controlStyle.stopIcon`

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Stoppen* Status des *Abspielen* Knopfs.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.22.141 chart.timeline.controlStyle.prevlIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Vorheriger* Knopf.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.22.142 chart.timeline.controlStyle.nextIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon für den *Nächster* Knopf.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.22.143 chart.timeline.controlStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Knopffarbe.

### 6.22.144 chart.timeline.controlStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#304654' ]

Farbe des Knopfrandes

## 6.22.145 chart.timeline.controlStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Knopfes.

## 6.22.146 chart.timeline.controlStyle.emphasis.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Knopfes im *hervorgehobenen* Zustand (wenn die Maus auf ihn zeigt).

## 6.22.147 chart.timeline.controlStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Knopffarbe.

### 6.22.148 chart.timeline.controlStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Farbe des Knopfrandes.

### 6.22.149 chart.timeline.controlStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Timeline
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

Breite des Knopfrandes.

## 6.23 ECharts (5) Title

### 6.23.1 chart.title.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titelkomponente, einschließlich Haupttitel und Untertitel.

In ECharts 2.x kann eine einzelne Instanz von Echarts maximal eine Titelkomponente enthalten. Von ECharts 3 kann es jedoch eine oder mehrere Titelkomponenten geben. Es ist möglich, dass alle benötigten Titel mehrerer Diagramme in einer Instanz definiert werden.

### 6.23.2 chart.title.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

### 6.23.3 chart.title.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Gibt an ob die Titelkomponente angezeigt wird.

### 6.23.4 chart.title.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Haupttitel-Text. Unterstützt \n für neue Zeilen.

### 6.23.5 chart.title.link

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: " ]

Der Hyperlink des Haupttiteltextes.

### 6.23.6 chart.title.target

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Öffnen des Hyperlinks des Haupttitels in dem angegebenen Tab. Optionen:

- 'self' öffnet es in der aktuellen Tab.
- 'blank' öffnet es in einem neuen Tab.

### 6.23.7 chart.title.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Textstil der Komponente.

### 6.23.8 chart.title.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#333" ]

Haupttitel Textfarbe.

### 6.23.9 chart.title.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Haupttitel Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 6.23.10 chart.title.textStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Haupttitel Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 6.23.11 chart.title.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Haupttitel Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.23.12 chart.title.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 18 ]

Haupttitel Schriftgröße.

### 6.23.13 chart.title.textStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

### 6.23.14 chart.title.textStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'top'
- 'middle'
- 'bottom'

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### 6.23.15 chart.title.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.23.16 chart.title.textStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, standardmäßig transparent.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 6.23.17 chart.title.textStyle.borderColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Titels. Unterstützt das gleiche Farbformat wie backgroundColor.

### 6.23.18 chart.title.textStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Titels.

### 6.23.19 chart.title.textStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die vier Ecken gesondert zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // Setze den gleichen Radius für alle vier Ecken
borderRadius: [5, 5, 0, 0] // (Im Uhrzeigersinn oben links, oben rechts, unten rechts
und unten links)
```

### 6.23.20 chart.title.textStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Innenabstand um das Textfragment, zum Beispiel:

- padding: [3, 4, 5, 6] steht für Innenabstand von [oben, rechts, unten, links].
- padding: 4 steht für padding: [4, 4, 4, 4].
- padding: [3, 4] steht für padding: [3, 4, 3, 4].

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts ohne Innenabstand spezifizieren.

### 6.23.21 chart.title.textStyle.shadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textblocks.

### 6.23.22 chart.title.textStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textblocks.

### 6.23.23 chart.title.textStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der X-Achse.

### 6.23.24 chart.title.textStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Textblock entlang der Y-Achse.

### 6.23.25 chart.title.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.23.26 chart.title.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.



Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.23.27 chart.title.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.23.28 chart.title.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.23.29 chart.title.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.23.30 chart.title.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.23.31 chart.title.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.23.32 chart.title.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.23.33 chart.title.textStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.23.34 chart.title.subtext

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "" ]

Untertiteltext. Unterstützt \n für neue Zeilen.

### 6.23.35 chart.title.sublink

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '' ]

Der Hyperlink des Untertiteltexes.

### 6.23.36 chart.title.subtarget

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'blank' ]

Öffnen des Hyperlinks des Untertitels in dem angegebenen Tab. Optionen:

- 'self' öffnet es in der aktuellen Tab.
- 'blank' öffnet es in einem neuen Tab.

### 6.23.37 chart.title.subtextStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Stil des Subtexts.

### 6.23.38 chart.title.subtextStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#aaa" ]

Untertitel Textfarbe.

### 6.23.39 chart.title.subtextStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Untertitel Schriftstil.

Mögliche Werte:

- 'normal'
- 'italic'
- 'oblique'

Erlaubte Werte

- normal
- italic
- oblique

### 6.23.40 chart.title.subtextStyle.fontWeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: normal ]

Untertitel Schriftstärke.

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### 6.23.41 chart.title.subtextStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Untertitel Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.23.42 chart.title.subtextStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Untertitel Schriftgröße.

### 6.23.43 chart.title.subtextStyle.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Horizontale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'left'
- 'center'
- 'right'

Wenn align im rich nicht gesetzt wird, wird das align der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  align: right,
  rich: {
    a: {
      // `align` ist nicht gesetzt, wird zu right
    }
  }
}
```

#### Erlaubte Werte

- left
- center
- right



### 6.23.44 chart.title.subtextStyle.verticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Vertikale Ausrichtung des Texts, standardmäßig automatisch.

Mögliche Werte:

- 'top'
- 'middle'
- 'bottom'

Wenn verticalAlign im rich nicht gesetzt wird, wird das verticalAlign der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  verticalAlign: bottom,
  rich: {
    a: {
      // `verticalAlign` ist nicht gesetzt, wird zu bottom
    }
  }
}
```

### 6.23.45 chart.title.subtextStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.23.46 chart.title.subtextStyle.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 6.23.47 chart.title.subtextStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Subtexts. Unterstützt das gleiche Farbformat wie backgroundColor.

### 6.23.48 chart.title.subtextStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Subtexts.

### 6.23.49 chart.title.subtextStyle.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 6.23.50 chart.title.subtextStyle.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Subtext um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.23.51 chart.title.subtextStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.23.52 chart.title.subtextStyle.shadowBlur

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 6.23.53 chart.title.subtextStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.23.54 chart.title.subtextStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

### 6.23.55 chart.title.subtextStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

#### Erlaubte Werte

- {percent}

### 6.23.56 chart.title.subtextStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.23.57 chart.title.subtextStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.23.58 chart.title.subtextStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.23.59 chart.title.subtextStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.23.60 chart.title.subtextStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.23.61 chart.title.subtextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.23.62 chart.title.subtextStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.23.63 chart.title.subtextStyle.rich

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Rich text styles" können in dieser rich-Eigenschaft definiert werden. Zum Beispiel:

```

label: {
  // Stile die in 'rich' definiert wurden können auf Textfragmente
  // durch das Hinzufügen von Markierungen angewendet werden, z.B.
  // `{styleName|text content text content}`.
  // `'\n'` ist der Zeilenumbruch.
  formatter: [
    '{a|Stil "a" wird auf dieses Fragment angewendet}'
    '{b|Stil "b" wird auf dieses Fragment angewendet}Dieses Fragment benutzt den
Standardstil{x|Benutzt Stil "x"}'
  ].join('\n'),

  rich: {
    a: {
      color: 'red',
      lineHeight: 10
    },
    b: {
      backgroundColor: {
        image: 'xxx/xxx.jpg'
      },
      height: 40
    },
    x: {
      fontSize: 18,
      fontFamily: 'Microsoft YaHei',
      borderColor: '#449933',
      borderRadius: 4
    },
    ...
  }
}

```

### 6.23.64 chart.title.triggerEvent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ob ein Event ausgelöst wird.

### 6.23.65 chart.title.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der Titel um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.23.66 chart.title.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Der Abstand zwischen dem Haupttitel und dem Untertitel.

### 6.23.67 chart.title.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente plziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel plziert.

### 6.23.68 chart.title.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.23.69 chart.title.left

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 6.23.70 chart.title.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.23.71 chart.title.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

## Erlaubte Werte

- {percent}
- {integer}

### 6.23.72 chart.title.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Grid-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

#### Erlaubte Werte

- {percent}
- {integer}

### 6.23.73 chart.title.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Hintergrundfarbe des Titels, welche standardmäßig transparent ist.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'.

### 6.23.74 chart.title.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe des Titels. Unterstützt das gleiche Farbformat wie backgroundColor.

### 6.23.75 chart.title.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite des Titels.

### 6.23.76 chart.title.borderRadius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Der Radius der abgerundeten Ecken. Die Einheit ist px. Unterstützt die Verwendung von Arrays, um die 4 Radien einzeln zu spezifizieren.

Zum Beispiel:

```
borderRadius: 5, // setze eine konsistente Größe für die 4 Ecken
borderRadius: [5, 5, 0, 0] // im Uhrzeigersinn oben links, oben rechts, unten rechts, unten links
```

### 6.23.77 chart.title.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title



<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist und backgroundColor einen anderen Wert als transparent hat.

### 6.23.78 chart.title.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.23.79 chart.title.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show: true konfiguriert ist.

### 6.23.80 chart.title.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

**Achtung:** Diese Eigenschaft funktioniert nur falls show:true konfiguriert ist.

### 6.23.81 chart.title.textStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.23.82 chart.title.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.23.83 chart.title.subtextStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen
- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.23.84 chart.title.subtextStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.23.85 chart.title.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die horizontale Ausrichtung der Komponente (einschließlich „Text“ und „Subtext“). Optionale Werte: 'auto', 'left', 'right', 'center'.

Erlaubte Werte

- auto
- left
- right
- center

### 6.23.86 chart.title.textVerticalAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Title
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Die vertikale Ausrichtung der Komponente (einschließlich „Text“ und „Subtext“). Optionale Werte: 'auto', 'top', 'bottom', 'middle'.

#### Erlaubte Werte

- auto
- top
- bottom
- middle

## 6.24 ECharts (5) Toolbox

### 6.24.1 chart.toolbox.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Komponenten-ID, nicht spezifiziert per Default. Wenn eine ID gesetzt wird, kann auf die Komponente per ID über die EChart-JavaScript-API zugegriffen werden.

### 6.24.2 chart.toolbox.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Zeigt die Tool-Box-Komponente an.

### 6.24.3 chart.toolbox.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'horizontal' ]

Die Layout-Orientierung der EChart-Toolbox Icons.

Erlaubte Werte

- horizontal
- vertical

### 6.24.4 chart.toolbox.itemSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 15 ]

Die Größe der EChart-Toolbox-Icons.

### 6.24.5 chart.toolbox.itemGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

Der Abstand zwischen den Toolbox-Icons. Bezieht sich je nach Layout auf den vertikalen oder horizontalen Abstand.

### 6.24.6 chart.toolbox.showTitle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Aktiviert die Titel-Text zu den Toolbox-Icons bei Mouse-Over.

true oder false

### 6.24.7 chart.toolbox.feature.saveAsImage

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Als Bild speichern.

### 6.24.8 chart.toolbox.feature.saveAsImage.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'png' ]

Format in dem das Bild gespeichert wird. Unterstützt 'png' und 'jpeg'.

### 6.24.9 chart.toolbox.feature.saveAsImage.name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Name mit dem das Bild gespeichert wird. Default-Wert ist title.text.

### 6.24.10 chart.toolbox.feature.saveAsImage.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]



Hintergrundfarbe mit der das Bild gespeichert wird. Default-Wert ist backgroundColor. Ist backgroundColor nicht gesetzt, wird die Farbe Weiß verwendet.

### 6.24.11 chart.toolbox.feature.saveAsImage.excludeComponents

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['toolbox'] ]

Komponenten die beim Export ausgeschlossen werden. Per default wird toolbox ausgeschlossen.

### 6.24.12 chart.toolbox.feature.saveAsImage.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 6.24.13 chart.toolbox.feature.saveAsImage.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'save as image' ]

#### 6.24.14 chart.toolbox.feature.saveAsImage.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 6.24.15 chart.toolbox.feature.saveAsImage.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellung des 'Als Bild speichern'-Icons.

#### 6.24.16 chart.toolbox.feature.saveAsImage.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.17 chart.toolbox.feature.saveAsImage.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.24.18 chart.toolbox.feature.saveAsImage.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.19 chart.toolbox.feature.saveAsImage.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.20 chart.toolbox.feature.saveAsImage.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.21 chart.toolbox.feature.saveAsImage.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.22 chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.23 chart.toolbox.feature.saveAsImage.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.24 chart.toolbox.feature.saveAsImage.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.24.25 chart.toolbox.feature.saveAsImage.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

#### 6.24.26 chart.toolbox.feature.saveAsImage.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

## Erlaubte Werte

- left
- right

## 6.24.27 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

## 6.24.28 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

## 6.24.29 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.30 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.31 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.



### 6.24.32 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.33 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.34 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.35 chart.toolbox.feature.saveAsImage.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.36 chart.toolbox.feature.saveAsImage.pixelRatio

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Auflösungsrate in der das Bild gespeichert wird, welche standardmäßig die des Containers ist. Werte größer als 1 (z.B. 2) werden unterstützt, falls eine höhere Auflösung benötigt wird.

### 6.24.37 chart.toolbox.feature.restore

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Einstellungen wiederherstellen"-Objekt

### 6.24.38 chart.toolbox.feature.restore.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 6.24.39 chart.toolbox.feature.restore.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'restore' ]

### 6.24.40 chart.toolbox.feature.restore.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 6.24.41 chart.toolbox.feature.restore.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Wiederherstellen"-Icons

#### 6.24.42 chart.toolbox.feature.restore.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

#### 6.24.43 chart.toolbox.feature.restore.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

#### 6.24.44 chart.toolbox.feature.restore.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

#### 6.24.45 chart.toolbox.feature.restore.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed

- dotted

#### 6.24.46 chart.toolbox.feature.restore.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 6.24.47 chart.toolbox.feature.restore.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 6.24.48 chart.toolbox.feature.restore.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 6.24.49 chart.toolbox.feature.restore.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 6.24.50 chart.toolbox.feature.restore.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.24.51 chart.toolbox.feature.restore.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Position der Beschriftung.

#### Erlaubte Werte

- left
- right
- top
- bottom

### 6.24.52 chart.toolbox.feature.restore.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

#### Erlaubte Werte

- left
- right

### 6.24.53 chart.toolbox.feature.restore.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]



Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

#### 6.24.54 chart.toolbox.feature.restore.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

#### 6.24.55 chart.toolbox.feature.restore.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

#### 6.24.56 chart.toolbox.feature.restore.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.57 chart.toolbox.feature.restore.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.58 chart.toolbox.feature.restore.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.59 chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.60 chart.toolbox.feature.restore.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.61 chart.toolbox.feature.restore.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.62 chart.toolbox.feature.dataView

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tool zur Dateneinsicht, welches Daten im momentanen Diagramm anzeigt und das Diagramm nach Änderungen erneuert.

### 6.24.63 chart.toolbox.feature.dataView.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 6.24.64 chart.toolbox.feature.dataView.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'data view' ]

### 6.24.65 chart.toolbox.feature.dataView.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.66 chart.toolbox.feature.dataView.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Dateneinsicht"-Icons

### 6.24.67 chart.toolbox.feature.dataView.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.68 chart.toolbox.feature.dataView.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.24.69 chart.toolbox.feature.dataView.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.70 chart.toolbox.feature.dataView.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.71 chart.toolbox.feature.dataView.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.72 chart.toolbox.feature.dataView.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.73 chart.toolbox.feature.dataView.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.74 chart.toolbox.feature.dataView.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.75 chart.toolbox.feature.dataView.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.76 chart.toolbox.feature.dataView.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 6.24.77 chart.toolbox.feature.dataView.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

## Erlaubte Werte

- left
- right

## 6.24.78 chart.toolbox.feature.dataView.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Symbolstil im hervorgehobenen Zustand.

## 6.24.79 chart.toolbox.feature.dataView.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

## 6.24.80 chart.toolbox.feature.dataView.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 6.24.81 chart.toolbox.feature.dataView.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.82 chart.toolbox.feature.dataView.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.83 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.84 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.85 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.24.86 chart.toolbox.feature.dataView.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

## 6.24.87 chart.toolbox.feature.dataView.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 6.24.88 chart.toolbox.feature.dataView.readOnly

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Ob es nur Lesezugriff erlaubt.

## 6.24.89 chart.toolbox.feature.dataView.optionToContent

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

```
(option:Object) => HTMLDomElement|string
```

Definiere eine Funktion, um `dataView` zu präsentieren. Sie wird verwendet, um die Standard-textarea für ergiebigere Datenverarbeitung zu ersetzen. Sie kann ein DOM-Objekt oder einen HTML-String zurückgeben.

Zum Beispiel:

```
optionToContent: function(opt) {
  var axisData = opt.xAxis[0].data;
  var series = opt.series;
  var table = '<table style="width:100%;text-align:center"><tbody><tr>'
    + '<td>Time:</td>'
    + '<td>' + series[0].name + '</td>'
    + '<td>' + series[1].name + '</td>'
    + '</tr>';
  for (var i = 0, l = axisData.length; i < l; i++) {
    table += '<tr>'
      + '<td>' + axisData[i] + '</td>'
      + '<td>' + series[0].data[i] + '</td>'
      + '<td>' + series[1].data[i] + '</td>'
      + '</tr>';
  }
  table += '</tbody></table>';
  return table;
}
```

## 6.24.90 chart.toolbox.feature.dataView.contentToOption

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

```
(container:HTMLDomElement, option:Object) => Object
```

Wenn optionToContent verwendet wird, falls die Aktualisierung von Diagrammen nach Datenveränderungen unterstützt werden soll, muss die Logik der Zusammenführung von Optionen in dieser Funktion implementiert werden.

### 6.24.91 chart.toolbox.feature.dataView.lang

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['data view', 'turn off', 'refresh'] ]

Es gibt 3 Namen im data view, diese sind 'data view', 'turn off' und 'refresh'.

### 6.24.92 chart.toolbox.feature.dataView.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Hintergrundfarbe der schwebenden Ebene im data view.

### 6.24.93 chart.toolbox.feature.dataView.textareaColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Hintergrundfarbe des Eingabebereichs der schwebenden Ebene im data view.

### 6.24.94 chart.toolbox.feature.dataView.textareaBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Rahmenfarbe des Eingabebereichs der schwebenden Ebene im data view.

### 6.24.95 chart.toolbox.feature.dataView.textColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]



Textfarbe.

### 6.24.96 chart.toolbox.feature.dataView.buttonColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#c23531' ]

Knopffarbe.

### 6.24.97 chart.toolbox.feature.dataView.buttonTextColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Farbe der Knopfbeschriftung.

### 6.24.98 chart.toolbox.feature.dataZoom

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Datenbereich-zooming, welches momentan nur rechteckige Koordinaten unterstützt.

### 6.24.99 chart.toolbox.feature.dataZoom.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das Tool angezeigt werden soll.

### 6.24.100 chart.toolbox.feature.dataZoom.title

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wiederhergestellter und gezoomter Titeltext.

### 6.24.101 chart.toolbox.feature.dataZoom.title.zoom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'area zooming' ]

### 6.24.102 chart.toolbox.feature.dataZoom.title.back

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'restore area zooming' ]

### 6.24.103 chart.toolbox.feature.dataZoom.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

"Zooming und Wiederherstellung"-Icon Pfad

### 6.24.104 chart.toolbox.feature.dataZoom.icon.zoom

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

## 6.24.105 chart.toolbox.feature.dataZoom.icon.back

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

## 6.24.106 chart.toolbox.feature.dataZoom.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Zoom auf Datenbereich"-Icons

### 6.24.107 chart.toolbox.feature.dataZoom.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.108 chart.toolbox.feature.dataZoom.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.24.109 chart.toolbox.feature.dataZoom.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.110 chart.toolbox.feature.dataZoom.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.111 chart.toolbox.feature.dataZoom.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.112 chart.toolbox.feature.dataZoom.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.113 chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.114 chart.toolbox.feature.dataZoom.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.115 chart.toolbox.feature.dataZoom.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.116 chart.toolbox.feature.dataZoom.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 6.24.117 chart.toolbox.feature.dataZoom.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 6.24.118 chart.toolbox.feature.dataZoom.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.119 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 6.24.120 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.121 chart.toolbox.feature.dataZoom.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.122 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.123 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.124 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.125 chart.toolbox.feature.dataZoom.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.126 chart.toolbox.feature.dataZoom.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.127 chart.toolbox.feature.dataZoom.xAxisIndex

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert welche xAxis kontrolliert werden sollen. Standardmäßig werden alle X-Achsen kontrolliert. Wird der Wert auf false gesetzt, werden keine X-Achsen kontrolliert. Wird der Wert auf 3 gesetzt, wird die X-Achse mit axisIndex von 3 kontrolliert. Wird der Wert auf [0, 3] gesetzt, werden die X-Achsen mit axisIndex von 0 und 3 kontrolliert.

### 6.24.128 chart.toolbox.feature.dataZoom.yAxisIndex

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiert welche yAxis kontrolliert werden sollen. Standardmäßig werden alle Y-Achsen kontrolliert. Wird der Wert auf false gesetzt, werden keine Y-Achsen kontrolliert. Wird der Wert auf 3 gesetzt, wird die Y-Achse mit axisIndex von 3 kontrolliert. Wird der Wert auf [0, 3] gesetzt, werden die Y-Achsen mit axisIndex von 0 und 3 kontrolliert.

### 6.24.129 chart.toolbox.feature.magicType

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Magic type wechsel. Beispiel:

```
feature: {
  magicType: {
    type: ['line', 'bar', 'stack', 'tiled']
  }
}
```

### 6.24.130 chart.toolbox.feature.magicType.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: true ]

Ob das tool angezeigt werden soll.

### 6.24.131 chart.toolbox.feature.magicType.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Aktivierte magic types. Möglich sind 'line' (für Liniendiagramme), 'bar' (für Balkendiagramme), 'stack' (für gestapelte Diagramme) und 'tiled' (für gekachelte Diagramme).

### 6.24.132 chart.toolbox.feature.magicType.title

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titel für verschiedene typen, können separat konfiguriert werden.

### 6.24.133 chart.toolbox.feature.magicType.title.line

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for line charts' ]

#### 6.24.134 chart.toolbox.feature.magicType.title.bar

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for bar charts' ]

#### 6.24.135 chart.toolbox.feature.magicType.title.stack

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for stacked charts' ]

#### 6.24.136 chart.toolbox.feature.magicType.title.tiled

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'for tiled charts' ]

### 6.24.137 chart.toolbox.feature.magicType.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die verschiedenen Typen von icon-Pfad, welche individuell konfiguriert werden können.

### 6.24.138 chart.toolbox.feature.magicType.icon.line

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.139 chart.toolbox.feature.magicType.icon.bar

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 6.24.140 chart.toolbox.feature.magicType.icon.stack

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

#### 6.24.141 chart.toolbox.feature.magicType.icon.tiled

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.142 chart.toolbox.feature.magicType.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des "Magic-Typ wechseln"-Icons

### 6.24.143 chart.toolbox.feature.magicType.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.144 chart.toolbox.feature.magicType.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.24.145 chart.toolbox.feature.magicType.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.146 chart.toolbox.feature.magicType.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

#### 6.24.147 chart.toolbox.feature.magicType.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

#### 6.24.148 chart.toolbox.feature.magicType.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

#### 6.24.149 chart.toolbox.feature.magicType.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

#### 6.24.150 chart.toolbox.feature.magicType.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

#### 6.24.151 chart.toolbox.feature.magicType.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

#### 6.24.152 chart.toolbox.feature.magicType.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 6.24.153 chart.toolbox.feature.magicType.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 6.24.154 chart.toolbox.feature.magicType.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

Symbolstil im hervorgehobenen Zustand.

### 6.24.155 chart.toolbox.feature.magicType.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimalformat verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.156 chart.toolbox.feature.magicType.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 6.24.157 chart.toolbox.feature.magicType.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.158 chart.toolbox.feature.magicType.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.159 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.



## 6.24.160 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

## 6.24.161 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

## 6.24.162 chart.toolbox.feature.magicType.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.163 chart.toolbox.feature.magicType.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.164 chart.toolbox.feature.magicType.option

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationsobjekt für jeden Typ, welche verwendet werden wenn zu einem bestimmten Typ gewechselt wird..

### 6.24.165 chart.toolbox.feature.magicType.option.line

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 6.24.166 chart.toolbox.feature.magicType.option.bar

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 6.24.167 chart.toolbox.feature.magicType.option.stack

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

### 6.24.168 chart.toolbox.feature.magicType.option.tiled

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 6.24.169 chart.toolbox.feature.magicType.seriesIndex

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Serienliste für jeden Typ.

## 6.24.170 chart.toolbox.feature.magicType.seriesIndex.line

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 6.24.171 chart.toolbox.feature.magicType.seriesIndex.bar

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 6.24.172 chart.toolbox.feature.magicType.seriesIndex.stack

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 6.24.173 chart.toolbox.feature.magicType.seriesIndex.tiled

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

TODO

## 6.24.174 chart.toolbox.feature.brush

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Pinsel-Auswählen Icon.

Kann auch bei brush.toolbox konfiguriert werden.

## 6.24.175 chart.toolbox.feature.brush.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob das tool angezeigt werden soll.

## 6.24.176 chart.toolbox.feature.brush.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Verwendete Icons, deren Werte sind:

- 'rect': Ermöglicht Auswahl mit rechteckigem Bereich.
- 'polygon': Ermöglicht Auswahl mit beliebiger Form.
- 'lineX': Ermöglicht horizontale Auswahl.
- 'lineY': Ermöglicht vertikale Auswahl.
- 'keep': Wechsel zwischen *einzelner Auswahl* und *multipler Auswahl*. Die Letztere kann mehrere Bereiche Auswählen während die Erstere vorherige Auswahlen abbricht.
- 'clear': Bricht alle Auswahlen ab.

## 6.24.177 chart.toolbox.feature.brush.icon

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Icon-Pfad für jedes Icon.

### 6.24.178 chart.toolbox.feature.brush.icon.rect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.179 chart.toolbox.feature.brush.icon.polygon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über

Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.180 chart.toolbox.feature.brush.icon.lineX

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.181 chart.toolbox.feature.brush.icon.lineY

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.



### 6.24.182 chart.toolbox.feature.brush.icon.keep

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.183 chart.toolbox.feature.brush.icon.clear

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Kann auf ein Bild gesetzt werden über 'image://url', wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.24.184 chart.toolbox.feature.brush.title

<b>Datentyp</b>	OBJECT
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Titel.

### 6.24.185 chart.toolbox.feature.brush.title.rect

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Rectangle selection' ]

### 6.24.186 chart.toolbox.feature.brush.title.polygon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Polygon selection' ]

### 6.24.187 chart.toolbox.feature.brush.title.lineX

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Horizontal selection' ]

### 6.24.188 chart.toolbox.feature.brush.title.lineY

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Vertical selection' ]

### 6.24.189 chart.toolbox.feature.brush.title.keep

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Keep previous selection' ]

### 6.24.190 chart.toolbox.feature.brush.title.clear

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'Clear selection' ]

### 6.24.191 chart.toolbox.iconStyle

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Stileinstellungen des geteilten Icons.

### 6.24.192 chart.toolbox.iconStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.193 chart.toolbox.iconStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #666 ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.24.194 chart.toolbox.iconStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.195 chart.toolbox.iconStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.196 chart.toolbox.iconStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.197 chart.toolbox.iconStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.198 chart.toolbox.iconStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.199 chart.toolbox.iconStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.200 chart.toolbox.iconStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.201 chart.toolbox.iconStyle.textPosition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Position der Beschriftung.

Erlaubte Werte

- left
- right
- top
- bottom

### 6.24.202 chart.toolbox.iconStyle.textAlign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ausrichtung des Beschriftungstextes.

Erlaubte Werte

- left
- right

### 6.24.203 chart.toolbox.iconStyle.emphasis

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

Symbolstil im hervorgehobenen Zustand.

### 6.24.204 chart.toolbox.iconStyle.emphasis.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch Hexadezimals Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

### 6.24.205 chart.toolbox.iconStyle.emphasis.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "#000" ]

Rahmenfarbe, deren Format ähnlich zu der von color ist.

### 6.24.206 chart.toolbox.iconStyle.emphasis.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Kein Rahmen wenn auf 0 gesetzt.

### 6.24.207 chart.toolbox.iconStyle.emphasis.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.24.208 chart.toolbox.iconStyle.emphasis.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente einzustellen.

### 6.24.209 chart.toolbox.iconStyle.emphasis.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt gleiches Format wie color.

### 6.24.210 chart.toolbox.iconStyle.emphasis.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.24.211 chart.toolbox.iconStyle.emphasis.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.24.212 chart.toolbox.iconStyle.emphasis.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.24.213 chart.toolbox.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.24.214 chart.toolbox.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 2 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.24.215 chart.toolbox.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 6.24.216 chart.toolbox.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

#### Erlaubte Werte

- {integer}
- {percent}
- top
- middle
- bottom

### 6.24.217 chart.toolbox.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

Standardmäßig adaptiv.

### 6.24.218 chart.toolbox.bottom

<b>Datentyp</b>	UNKNOWN
-----------------	---------

<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der Toolbox-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

Standardmäßig adaptiv.

### 6.24.219 chart.toolbox.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Breite der Toolbox-Komponente. Standardmäßig adaptiv.

### 6.24.220 chart.toolbox.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Höhe der Toolbox-Komponente. Standardmäßig adaptiv.

### 6.24.221 chart.toolbox.defaulticons

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Spezifiziert, ob die standardmäßigen Icons für die Toolbox verwendet werden sollen, oder die von Tradui entworfenen Icons.

### 6.24.222 chart.toolbox.feature.myToolToggleSeriesVisibility.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob die Toolbox-Option 'ToggleVisibility' angezeigt werden soll. Wird dieser Knopf angeklickt, so werden alle Legendeninformationen sichtbar oder unsichtbar

### 6.24.223 chart.toolbox.feature.myToolToggleSeriesVisibility\_RescaleYAxis.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Ob die Toolbox-Option für 'ToggleVisibility' angezeigt werden soll. Wird dieser Knopf angeklickt, so werden alle Legendeninformationen sichtbar oder unsichtbar. Zusätzlich kann die Skalierung der Y-Achse durch Setzen des Wertes mit der Option "`chart.toolbox.feature.Rescale Axis.function`" angepasst werden.

### 6.24.224 chart.toolbox.feature.RescaleYAxis.function

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Skalierung der Y-Achse kann mit dieser Option für die Toolbox-Option 'ToggleVisibility' angepasst werden.

### 6.24.225 chart.toolbox.feature.myToolRestore.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Gibt an, ob die benutzerdefinierte Wiederherstellungsoption angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option `Echarts.connect ()` verbunden werden.

### 6.24.226 chart.toolbox.feature.myToolBarChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte BarChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 6.24.227 chart.toolbox.feature.myToolLineChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte LineChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

### 6.24.228 chart.toolbox.feature.myToolStackedChart.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<span>TESTED</span>
<b>Verfügbar seit Version</b>	2.3.5

### Schlagworte / Tags

[ Default-Wert: false]

Gibt an, ob die benutzerdefinierte StackedChart angezeigt werden soll. Diese Option wird verwendet, wenn die Diagramme mit der Option Echarts.connect () verbunden werden.

## 6.24.229 chart.toolbox.feature.myToolRestore.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die Restore-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
<span style="font-size:10.0pt">dashlet.addOption("chart.toolbox.feature.myToolRestore.additionalscript", "console.log('here');");</span>
```

## 6.24.230 chart.toolbox.feature.myToolBarChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die BarChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolBarChart.additionalscript",
"console.log('here');");
```

### 6.24.231 chart.toolbox.feature.myToolLineChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die LineChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolLineChart.additionalscript",
"console.log('here');");
```

### 6.24.232 chart.toolbox.feature.myToolStackedChart.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die StackedChart-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption("chart.toolbox.feature.myToolStackedChart.additionalscript",
"console.log('here');");
```

### 6.24.233 chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die ToggleVisibility-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption(
  "chart.toolbox.feature.myToolToggleSeriesVisibility.additionalscript",
  "console.log('here');");
```

### 6.24.234 chart.toolbox.feature.myToolToggleSeriesVisibility\_RescaleYAxis.additionalscript

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Toolbox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.13
<b>Schlagworte / Tags</b>	

Ein customized Javascript-Code kann für die ToggleVisibility-button gerendert werden. Diese Option ist zu verwenden, wenn die Diagramme mit der Option Echarts.connect() verbunden sind.

Beispiel:

```
dashlet.addOption(
  "chart.toolbox.feature.myToolToggleSeriesVisibility_RescaleYAxis.additionalscript",
  "console.log('here');");
```

## 6.25 ECharts (5) Tooltip

### 6.25.1 chart.tooltip.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Tooltip Komponente.

#### Allgemeine Einführung:

Tooltips können an verschiedenen Stellen konfiguriert werden:

- Global konfiguriert: tooltip
- In einem Koordinatensystem konfiguriert: grid.tooltip, polar.tooltip, single.tooltip
- In einer Serie konfiguriert: series.tooltip
- In jedem Element von series.data konfiguriert: series.data.tooltip

### 6.25.2 chart.tooltip.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Tooltip-Komponente angezeigt werden soll, beinhaltet schwebenden Layer des Tooltips und axisPointer.

### 6.25.3 chart.tooltip.trigger

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'item' ]

Form des Auslösens.

Mögliche Werte:

- 'item'  
Durch Datenelemente ausgelöst, was hauptsächlich für Diagramme ohne Kategorieachse, wie Streu- oder Kuchendiagramme, verwendet wird.
- 'axis'  
Durch Achsen ausgelöst, was hauptsächlich für Diagramme mit Kategorieachsen, wie Balken- oder Liniendiagramme, verwendet wird.  
ECharts 2.x unterstützt das Auslösen durch Achsen nur für Kategorieachsen. In ECharts 3 ist es für alle Achsentypen in grid und polar unterstützt. Weiterhin lassen sich Achsen über axisPointer.axis zuweisen.
- 'none'  
Durch nichts ausgelöst.

Erlaubte Werte

- item
- axis
- none

### 6.25.4 chart.tooltip.axisPointer.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Konfigurationselement für den Achsenindikator.

tooltip.axisPointer ist wie syntaktischer Zucker von axisPointer-Einstellungen an Achsen (zum Beispiel xAxis.axisPointer oder angleAxis.axisPointer). Detailliertere Einstellungen können an someAxis.axisPointer vorgenommen werden. Im Normalfall ist die Verwendung von tooltip.axisPointer jedoch praktischer.

**Hinweis:** Konfigurationen an tooltip.axisPointer haben eine niedrigere Priorität als someAxis.axisPointer

### 6.25.5 chart.tooltip.axisPointer.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'line' ]

Indikatorotyp.

Mögliche Werte:

- 'line': Linienindikator
- 'shadow': Schattierter Fadenkreuzindikator
- 'none': Kein Indikator angezeigt
- 'cross': Fadenkreuzindikator, welcher tatsächlich die Kurzfassung davon ist, zwei axisPointers auf zwei orthometrischen Achsen zu aktivieren.

Erlaubte Werte

- line
- shadow
- none
- cross

### 6.25.6 chart.tooltip.axisPointer.axis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: 'auto' ]

Die Koordinatenachse, welche 'x', 'y', 'radius' oder 'angle' sein kann. Standardmäßig wählt jedes Koordinatensystem automatisch die Achsen, welche den axisPointer anzeigen sollen (standardmäßig wird die Kategorieachse oder Zeitachse verwendet).

### Erlaubte Werte

- auto
- x
- y
- radius
- angle

## 6.25.7 chart.tooltip.axisPointer.snap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob automatisch auf Punkten eingerastet werden soll. Der Standardwert wird automatisch bestimmt.

Diese Funktion ergibt gewöhnlich für Wert- und Zeitachsen Sinn, wo kleine Punkte automatisch gesucht werden können.

## 6.25.8 chart.tooltip.axisPointer.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

z-Wert, welcher die Reihenfolge bestimmt, in der grafische Komponenten gezeichnet werden. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten übermalt werden.

### 6.25.9 chart.tooltip.axisPointer.label.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Label des axisPointers.

### 6.25.10 chart.tooltip.axisPointer.label.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Label angezeigt werden soll. Standardmäßig wird das Label nicht angezeigt. Ist jedoch tooltip.axisPointer auf 'cross' gesetzt, so wird das Label automatisch angezeigt.

### 6.25.11 chart.tooltip.axisPointer.label.precision

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Die Präzision des Wertes im Label. Wird standardmäßig automatisch bestimmt. Es lässt sich auch auf '2' setzen, was darauf hinweist, dass zwei Dezimalstellen reserviert werden.

## 6.25.12 chart.tooltip.axisPointer.label.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Der Formatierer des Labels.

Als String gesetzt wäre ein Beispiel: 'ein Text {value} ein Text', wobei {value} automatisch durch den Achsenwert ersetzt wird.

Als Funktion gesetzt:

### **Parameter:**

{Object} params: Enthält die folgenden Felder:

{Object} params.value: Aktueller Wert dieser Achse. Ist axis.type 'category', so ist es einer der Werte in axis.data. Ist axis.type 'time', so ist es ein Zeistempel.

{Array.<Object>} params.seriesData: Ein Array, welches Informationen über die nächstgelegenen Punkte enthält. jeder Eintrag ist:

```
{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}
```

Jeder Eintrag enthält zusätzlich Achseninformationen:

```
{
  axisDim: 'x', // 'x', 'y', 'angle', 'radius', 'single'
  axisId: 'xxx',
  axisName: 'xxx',
  axisIndex: 3,
  axisValue: 121, // Der momentane Wert von axisPointer
  axisValueLabel: 'text of value'
}
```

#### Rückgabewert:

Deer String der angezeigt werden soll..

Zum Beispiel:

```
formatter: function (params) {
  // Wenn axis.type 'time' ist
  return 'ein text' + echarts.format.formatTime(params.value);
}
```

### 6.25.13 chart.tooltip.axisPointer.label.margin

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Abstand zwischen Label und Achse.

### 6.25.14 chart.tooltip.axisPointer.label.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

### 6.25.15 chart.tooltip.axisPointer.label.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

#### Erlaubte Werte

- normal

- italic
- oblique

### 6.25.16 chart.tooltip.axisPointer.label.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.25.17 chart.tooltip.axisPointer.label.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.25.18 chart.tooltip.axisPointer.label.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße

### 6.25.19 chart.tooltip.axisPointer.label.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments.

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

## 6.25.20 chart.tooltip.axisPointer.label.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.25.21 chart.tooltip.axisPointer.label.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

## 6.25.22 chart.tooltip.axisPointer.label.textBorderColor

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.25.23 chart.tooltip.axisPointer.label.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.25.24 chart.tooltip.axisPointer.label.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.25.25 chart.tooltip.axisPointer.label.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.25.26 chart.tooltip.axisPointer.label.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.25.27 chart.tooltip.axisPointer.label.textShadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.25.28 chart.tooltip.axisPointer.label.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im axisPointer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.25.29 chart.tooltip.axisPointer.label.backgroundColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'auto' ]

Hintergrundfarbe des Labels, standardmäßig die gleiche wie axis.axisLine.lineStyle.color.

### 6.25.30 chart.tooltip.axisPointer.label.borderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Rahmenfarbe des Labels.

### 6.25.31 chart.tooltip.axisPointer.label.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Labels.

### 6.25.32 chart.tooltip.axisPointer.label.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 3 ]

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.25.33 chart.tooltip.axisPointer.label.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #aaa ]

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.25.34 chart.tooltip.axisPointer.label.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.25.35 chart.tooltip.axisPointer.label.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.25.36 chart.tooltip.axisPointer.lineStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'line' ist.

### 6.25.37 chart.tooltip.axisPointer.lineStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimalen Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

### 6.25.38 chart.tooltip.axisPointer.lineStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 1 ]

Linienbreite

### 6.25.39 chart.tooltip.axisPointer.lineStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Linientyp

Erlaubte Werte

- solid
- dashed
- dotted

### 6.25.40 chart.tooltip.axisPointer.lineStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content



<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

#### 6.25.41 chart.tooltip.axisPointer.lineStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

#### 6.25.42 chart.tooltip.axisPointer.lineStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.25.43 chart.tooltip.axisPointer.lineStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.25.44 chart.tooltip.axisPointer.lineStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.25.45 chart.tooltip.axisPointer.shadowStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

**Schlagworte / Tags**

Ist gültig wenn axisPointer.type 'shadow' ist.

### 6.25.46 chart.tooltip.axisPointer.shadowStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(150,150,150,0.3)' ]

Füllfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 6.25.47 chart.tooltip.axisPointer.shadowStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.25.48 chart.tooltip.axisPointer.shadowStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.25.49 chart.tooltip.axisPointer.shadowStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.25.50 chart.tooltip.axisPointer.shadowStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.25.51 chart.tooltip.axisPointer.shadowStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.25.52 chart.tooltip.axisPointer.crossStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ist gültig wenn axisPointer.type 'cross' ist.

### 6.25.53 chart.tooltip.axisPointer.crossStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: #555 ]

Linienfarbe.

Farbe kann in RGB dargestellt werden, zum Beispiel 'rgb(128, 128, 128)'. RGBA kann verwendet werden, falls ein Alpha-Kanal benötigt wird, zum Beispiel 'rgba(128, 128, 128, 0.5)'. Es lässt sich auch hexadezimaler Format verwenden, zum Beispiel '#ccc'. Neben einzelnen Farben werden auch Gradienten und Texturen unterstützt.

```
// Linearer Gradient. Die ersten vier Parameter sind x, y, x2 und y2, welche Werte aus dem Bereich 0 bis 1 haben und für prozentuale Positionen innerhalb der Bounding Box stehen. Wurde globalCoord auf 'true' gesetzt, so sind die ersten vier Parameter absolute Pixelwerte.
```

```
color: {
  type: 'linear',
  x: 0,
  y: 0,
  x2: 0,
  y2: 1,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Radialer Gradient. Die ersten drei Parameter sind x und y Position und Radius, ähnlich zum linearen Gradienten.
```

```
color: {
  type: 'radial',
  x: 0.5,
  y: 0.5,
  r: 0.5,
  colorStops: [{
    offset: 0, color: 'red' // Farbe bei 0% Position
  }, {
    offset: 1, color: 'blue' // Farbe bei 100% position
  }],
  globalCoord: false // Standardmäßig false
}
```

```
// Fill with texture
```

```
color: {
  image: imageDom, // HTMLImageElement und HTMLCanvasElement werden unterstützt, String Pfade werden nicht unterstützt.
  repeat: 'repeat' // Spezifiziert, ob die Textur sich wiederholen soll. Kann auf 'repeat-x', 'repeat-y' oder 'no-repeat' gesetzt werden.
}
```

## 6.25.54 chart.tooltip.axisPointer.crossStyle.width

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5



<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 1 ]

Linienbreite.

### 6.25.55 chart.tooltip.axisPointer.crossStyle.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'dashed' ]

Linientyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.25.56 chart.tooltip.axisPointer.crossStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.25.57 chart.tooltip.axisPointer.crossStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.25.58 chart.tooltip.axisPointer.crossStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.25.59 chart.tooltip.axisPointer.crossStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.25.60 chart.tooltip.axisPointer.crossStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.25.61 chart.tooltip.showContent

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob der schwebende Tooltip-Layer angezeigt werden soll. Dieser Wert sollte auf false gesetzt werden, falls der Tooltip nur sein Event auslösen oder den axisPointer ohne Inhalt anzeigen soll.

### 6.25.62 chart.tooltip.alwaysShowContent

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Inhalt des Tooltips durchgehend angezeigt werden soll. Standardmäßig wird er nach einiger Zeit verborgen. Dieser Wert kann auf true gesetzt werden, um das Anzeigen zu bewahren.

Dieses Attribut wurde zu ECharts 3.0 neu hinzugefügt.

### 6.25.63 chart.tooltip.triggerOn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'mousemove|click' ]

Konditionen unter denen der Tooltip auslöst. Mögliche Werte:

- 'mousemove'  
Löse aus wenn die Maus sich bewegt.
- 'click'  
Löse aus wenn die Maus klickt.
- 'mousemove|click'  
Löse aus wenn die Maus klickt und sich bewegt.
- 'none'  
Löse nicht durch bewegen und klicken der Maus aus. Der Tooltip kann manuell ausgelöst und verborgen werden indem action.tooltip.showTip und action.tooltip.hideTip aufgerufen werden. In diesem Fall kann er auch durch axisPointer.handle ausgelöst werden.

#### Erlaubte Werte

- mousemove
- click
- none
- mousemove|click

### 6.25.64 chart.tooltip.showDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Zeitverzögerung in ms bis der Tooltip angezeigt wird. Standardmäßig gibt es keine Verzögerung, und das setzen einer solchen ist nicht empfohlen. Nur gültig wenn triggerOn auf 'mousemove' gesetzt ist.

### 6.25.65 chart.tooltip.hideDelay

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 100 ]

Verzögerung in ms bis der Tooltip verborgen wird. Ist ungültig wenn alwaysShowContent auf true gesetzt ist.

### 6.25.66 chart.tooltip.enterable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob die Maus den schwebenden Tooltip-Layer betreten darf. Muss mit Inhalten des Tooltips, wie Links oder Schaltern, interagiert werden, so kann dieser Wert auf true gesetzt werden.

### 6.25.67 chart.tooltip.renderMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'html' ]

Rendermodus für den Tooltip. Steht standardmäßig auf 'html', wodurch ein zusätzliches DOM-Element für den Tooltip verwendet wird. Kann auch auf 'richText' gesetzt werden, wodurch der Tooltip im Canvas-Element gerendert wird (SVG rich text ist noch nicht implementiert). Dies ist sehr nützlich für Umgebungen die nicht über DOM verfügen, wie zum Beispiel Wechat Applikationen.

#### Erlaubte Werte

- html
- richText

### 6.25.68 chart.tooltip.confine

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob der Inhalt des Tooltips durch das rechteckige Fenster des Diagramms begrenzt werden soll.

Nützlich wenn der Tooltip abgeschnitten wird weil 'overflow: hidden' auf dem äußeren DOM der Diagramm-Instanz gesetzt ist oder wegen kleinen Bildschirmen auf Mobilgeräten.

## 6.25.69 chart.tooltip.transitionDuration

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0.4 ]

Die Übergangsdauer der Animation des Tooltips in Sekunden. Wird der Wert auf 0 gesetzt, so bewegt sich der Tooltip eng mit der Maus.

## 6.25.70 chart.tooltip.position

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Position des schwebenden Tooltip-Layers, welcher standardmäßig der Maus folgt.

Möglichkeiten:

- Array  
Stelle die Position des schwebenden Tooltip-Layers als Array dar, welches absolute Positionen und relative Prozentwerte unterstützt.  
Beispiel:

```
// absolute Position, welche 10px Abstand zur linken und 10px Abstand zur
// oberen Seite des Containers hat.
position: [10, 10]
// relative Position, genau in der Mitte des Containers
position: ['50%', '50%']
```

- Funktion  
Callback-Funktion der Form:

```
(point: Array, params: Object|Array.<Object>, dom: HTMLElement, rect: Object, size: Object) => Array
```

**Parameter:**

point: Position der Maus.

param: Wie im Formatierer.

dom: Das DOM-Objekt des Tooltips.

rect: Nur gültig wenn die Maus auf einem grafischen Element ist, steht für eine Bounding-Box mit x, y, width und height.

size: Die Größe des DOM ECharts-Containers. Zum Beispiel: {contentSize: [width, height], viewSize: [width, height]}.

**Rückgabe:**

Der Rückgabewert ist ein Array, welches die Position des Tooltips angibt und dafür absolute Pixel oder relative Prozentwerte verwenden kann.

Oder der Rückgabewert ist ein Objekt, wie {left: 10, top: 30}, or {right: '20%', bottom: 40}.

Zum Beispiel:

```
position: function (point, params, dom, rect, size) {
  // Oben festgelegt
  return [point[0], '10%'];
}
```

Oder:

```
position: function (pos, params, dom, rect, size) {
  // Tooltip wird rechts festgelegt wenn die Maus links ist,
  // und links wenn sie rechts ist.
  var obj = {top: 60};
  obj[['left', 'right'][+(pos[0] < size.viewSize[0] / 2)]] = 5;
  return obj;
}
```

- 'inside'  
Zentrale Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'top'  
Obere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'left'  
Linke Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'right'  
Rechte Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.
- 'bottom'  
Untere Position des grafischen Elements in dem sich die Maus befindet, ist nur gültig wenn der Auslöser 'item' ist.

## 6.25.71 chart.tooltip.formatter

**Datentyp**

UNKNOWN



<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Inhaltsformatierer des schwebenden Tooltip-Layers, welcher String-Templates und Callback-Funktionen unterstützt.

### 1. String-Template

Die Template-Variablen sind {a}, {b}, {c}, {d} und {e}, was für Serienname, Datenname, Datenwert und etc. steht. Wenn der Auslöser auf 'axis' gesetzt ist, können die Daten mehrerer Serien zur Verfügung stehen. In diesem Fall kann der Serienindex referenziert werden als {a0}, {a1}, oder {a2}.

{a}, {b}, {c}, {d} haben für unterschiedliche Serientypen unterschiedliche Bedeutungen:

- Linien- und Flächendiagramme, Balken- und Säulendiagramme, K-Diagramme: {a} für Serienname, {b} für Kategorienname, {c} für Datenwert, {d} für nichts;
- Streu- oder Blasendiagramm: {a} für Serienname, {b} für Datenname, {c} für Datenwert, {d} für nichts;
- Map: {a} for series name, {b} für Bereichsname, {c} für Datenvereinigung, {d} für nichts;
- Kuchendiagramm, Messdiagramm, Trichterdiagramm: {a} für Serienname, {b} für Name von Datenelement, {c} für Datenwert, {d} für Prozentwert.

### Beispiel:

```
formatter: '{b0}: {c0}<br />{b1}: {c1}'
```

### 2. Callback-Funktion

Das Format der Callback-Funktion:

```
(params: Object|Array, ticket: string, callback: (ticket: string, html: string)) => string
```

Der erste Parameter params ist der Datensatz den der Formatierer braucht. Sein Format ist wie folgt:

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,

  // Der Prozentanteil des Kuchendiagramms
  percent: number,
}

```

Wenn der Auslöser 'axis' ist oder der Tooltip von axisPointer ausgelöst wird, ist params das Datenarray mehrerer Serien. Der Inhalt jedes Elements des Arrays ist der gleiche wie oben. Bis auf,

```

{
  componentType: 'series',
  // Serientyp
  seriesType: string,
  // Serienindex in option.series
  seriesIndex: number,
  // Seriename
  seriesName: string,
  // Daten- oder Kategorienname
  name: string,
  // Datenindex im eingegebenen Datenarray
  dataIndex: number,
  // Originaldaten als Eingabe
  data: Object,
  // Wert der Daten
  value: number|Array,
  // Farbe der Daten
  color: string,
}

```

**Hinweis:** Ein Array zu benutzen um alle Parameter in ECharts 2.x darzustellen ist nicht länger unterstützt.

Der zweite Parameter ticket ist eine asynchrone Callback-Flagge welche zusammen mit dem dritten Parameter callback verwendet werden sollte.

Der dritte Parameter callback ist ein asynchroner Callback. Wird der Inhalt von Tooltip asynchron beschafft, so können ticket und html wie oben beschrieben verwendet werden um den Tooltip mit callback zu aktualisieren.

Beispiel:

```
formatter: function (params, ticket, callback) {
  $.get('detail?name=' + params.name, function (content) {
    callback(ticket, toHTML(content));
  });
  return 'Laden';
}
```

## 6.25.72 chart.tooltip.backgroundColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(50,50,50,0.7)' ]

Die Hintergrundfarbe des schwebenden Tooltip-Layers.

## 6.25.73 chart.tooltip.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#333' ]

Die Rahmenfarbe des schwebenden Tooltip-Layers.

## 6.25.74 chart.tooltip.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die Rahmenbreite des schwebenden Tooltip-Layers.

## 6.25.75 chart.tooltip.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand im Tooltip-Layer um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

## 6.25.76 chart.tooltip.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Textstil des schwebenden Tooltip-Layers.

## 6.25.77 chart.tooltip.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#fff' ]

Textfarbe.

## 6.25.78 chart.tooltip.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.25.79 chart.tooltip.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'
- 'lighter'
- 100 | 200 | 300 | 400...

### Erlaubte Werte

- normal
- bold
- bolder
- lighter

## 6.25.80 chart.tooltip.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: 'sans-serif' ]

Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.25.81 chart.tooltip.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 14 ]

Schriftgröße.

### 6.25.82 chart.tooltip.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```

{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}

```

### 6.25.83 chart.tooltip.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.25.84 chart.tooltip.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.25.85 chart.tooltip.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.25.86 chart.tooltip.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.25.87 chart.tooltip.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.25.88 chart.tooltip.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.25.89 chart.tooltip.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.25.90 chart.tooltip.textStyle.textShadowOffsetY

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.25.91 chart.tooltip.extraCssText

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zusätzlicher CSS-Stil für den schwebenden Layer. Das folgende ist ein Beispiel für das Hinzufügen eines Schattens.

```
extraCssText: 'box-shadow: 0 0 3px rgba(0, 0, 0, 0.3);'
```

### 6.25.92 chart.tooltip.axisPointer.label.textBorderStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

Erlaubte Werte

- dashed
- dotted
- solid

### 6.25.93 chart.tooltip.axisPointer.label.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderType` können wir den Linienstil flexibler gestalten.

### 6.25.94 chart.tooltip.axisPointer.label.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen

- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.25.95 chart.tooltip.axisPointer.label.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.25.96 chart.tooltip.axisPointer.lineStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

### 6.25.97 chart.tooltip.axisPointer.lineStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

## 6.25.98 chart.tooltip.axisPointer.lineStyle.join

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

## Erlaubte Werte

- bevel
- round
- miter

### 6.25.99 chart.tooltip.axisPointer.lineStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.25.100 chart.tooltip.axisPointer.crossStyle.miterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.25.101 chart.tooltip.axisPointer.crossStyle.join

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.25.102 chart.tooltip.axisPointer.crossStyle.cap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.



- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

## Erlaubte Werte

- butt
- round
- square

### 6.25.103 chart.tooltip.axisPointer.crossStyle.dashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Tooltip
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit Type können wir den Linienstil flexibler gestalten.

## 6.26 ECharts (5) Visual Map

### 6.26.1 chart.visualMap.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

visualMap ist eine Art Komponente für visuelles Encoding, welche die Daten auf visuelle Kanäle mappt. Zu diesen Kanälen zählt:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL.

- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbton in HSL.

Mehrere visualMap-Komponenten können in einer Diagramm-Instanz definiert werden, wodurch verschiedene Dimensionen einer Datenserie auf verschiedene visuelle Kanäle gemappt werden können.

visualMap kann als stetig oder stückweise definiert werden, was über die Eigenschaft type geschieht. Zum Beispiel:

```
option = {
  visualMap: [
    { // Die erste visualMap-Komponente
      type: 'continuous', // Definiert als stetige visualMap
      ...
    },
    { // Die zweite visualMap-Komponente
      type: 'piecewise', // Definiert als stückweise visualMap
      ...
    }
  ],
  ...
};
```

### Configure mapping

Die Dimension von series.data kann mit visualMap.dimension spezifiziert werden, von wo aus der Wert geholt und auf visuelle Kanäle gemappt werden kann. Diese Kanäle können in visualMap.inRange und visualMap.outOfRange definiert werden.

Soll in einer von einer visualMap kontrollierten Serie ein Element dieser Kontrolle entkommen, kann es wie folgt gesetzt werden:

```
series: {
  type: '...',
  data: [
    {name: 'Shanghai', value: 251},
    {name: 'Haikou', value: 21},
    // Markieren als `visualMap: false`, wodurch dieses Element nicht mehr von
    // visualMap kontrolliert wird
    // und die visuelle Konfiguration der Serie (wie color, symbol, ...) auf das
    // Element angewendet werden kann.
    {name: 'Beijing', value: 821, visualMap: false},
    ...
  ]
}
```

### Die Beziehung zwischen visualMap von ECharts3 und dataRange von ECharts2

dataRange aus ECharts2 wurde zu visualMap umbenannt, und das Ausmaß der Funktionalitäten wurde stark erweitert. Die Konfigurationen von dataRange sind kompatibel mit ECharts3, wo diese automatisch zu visualMap umgewandelt werden. Es wird empfohlen, in ECharts 3 visualMap statt dataRange zu verwenden.

## 6.26.2 chart.visualMap.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den Typ der visualMap.

Erlaubte Werte

- continuous
- piecewise

## 6.26.3 chart.visualMap.id

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

ID der Komponente, standardmäßig nicht definiert. Falls definiert, kann sie verwendet werden, um sich in Optionen oder API auf die Komponente zu beziehen.

## 6.26.4 chart.visualMap.min

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 0 ]

Spezifiziert den minimalen Datenwert für die visualMap-Komponente. [visualMap.min, visualMap.max] stellen den Bereich des visuellen Mappings dar.

Es ist anzumerken, dass min und max explizit spezifiziert werden sollten, und standardmäßig [0, 200] sind, nicht etwa dataMin und dataMax aus series.data.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) müssen max und min nicht spezifiziert werden.

## 6.26.5 chart.visualMap.max

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 200 ]

Spezifiziert den maximalen Datenwert für die visualMap-Komponente. [visualMap.min, visualMap.max] stellen den Bereich des visuellen Mappings dar.

Es ist anzumerken, dass min und max explizit spezifiziert werden sollten, und standardmäßig [0, 200] sind, nicht etwa dataMin und dataMax aus series.data.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) müssen max und min nicht spezifiziert werden.

## 6.26.6 chart.visualMap.range

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert den ausgewählten Bereich, also die Datenwerte, die zu den beiden Handles gehören. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    min: 0,
    max: 100,
    // Datenwerte die zu den beiden Handles gehören.
    range: [4, 15],
    ...
  }
});
```

### Automatische Anpassung wenn min oder max durch setOption verändert wird

- Wenn range nicht gesetzt ist (oder auf null oder undefined gesetzt ist)

Zum Beispiel:

```
chart.setOption({visualMap: {min: 10, max: 300}}); // range ist nicht gesetzt, wird
standardmäßig [min, max], also [10, 300].
```

```
chart.setOption({visualMap: {min: 0, max: 400}}); // Modifiziere min und max wieder
mit setOption.
// range wird dann automatisch modifiziert auf das neue [min, max], also [0, 400].
```

- Wenn range explizit gesetzt ist, zum Beispiel auf [10, 300]

Zum Beispiel:

```
chart.setOption({visualMap: {min: 10, max: 300, range: [20, 80]}}); // range wurde
auf [20, 80] gesetzt.
```

```
chart.setOption({visualMap: {min: 0, max: 400}}); // Modifiziere min und max wieder
mit setOption.
// range behält den originalen Wert ([20, 80]) und passt sich nicht automatisch an.
```

```
chart.setOption({visualMap: {range: null}}); // Setze range auf null
// Die automatische Anpassung von range aktiviert sich und setzt range auf [min,
max], also [0, 400].
```

Die range die von getOption zurückgegeben wird ist immer ein Array, aber nie null oder undefined.

## 6.26.7 chart.visualMap.calculable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: false ]

Spezifiziert, ob Handles angezeigt werden sollen, welche gezogen werden können um den "ausgewählten Bereich" zu verändern.

Anmerkung: Um mit ECharts 2 kompatibel zu bleiben, bleibt die folgende, seltsam erscheinende Regel erhalten: Wenn `visualMap.type` nicht gesetzt ist und `visualMap.calculable` auf `true` gesetzt ist, wird `visualMap.type` automatisch auf 'continuous' gesetzt. Einstellungen wie `visualMap.splitNumber` werden dabei ignoriert. Daher wird empfohlen, `visualMap.type` explizit zu setzen, um Unklarheiten vorzubeugen.

## 6.26.8 chart.visualMap.realtime

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die Ansicht in Echtzeit aktualisiert werden soll, wenn ein Handle gezogen wird.

- Für `true` wird die Ansicht in Echtzeit beim Ziehen aktualisiert.
- Für `false` wird die Ansicht aktualisiert, wenn das Ziehen beendet wurde.

## 6.26.9 chart.visualMap.inverse

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Spezifiziert, ob das Layout der `visualMap`-Komponente umgekehrt werden soll.

**Für stetige visualMap:**

Steht inverse auf false, so ist die Richtung des Layouts wie im kartesischen Koordinatensystem. Das heißt:

- Steht visualMap.orient auf 'vertical', so werden große Datenwerte oben und kleine Datenwerte unten platziert.
- Steht visualMap.orient auf 'horizontal', so werden große Datenwerte rechts und kleine Datenwerte links platziert.

#### Für stückweise visualMap:

Im **CONTINUOUS-AVERAGE** Modus (wenn visualMap.splitNumber benutzt wird) ist die Regel des Layouts identisch zur stetigen visualMap.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) wird das Layout der Elemente durch ihre Reihenfolge in der Definition der Stücke bzw. Kategorien bestimmt. Das heißt:

- Steht inverse auf false:
  - Wenn visualMap.orient auf 'vertical' steht, wird pieces[0] bzw. categories[0] oben platziert.
  - Wenn visualMap.orient auf 'horizontal' steht, wird pieces[0] bzw. categories[0] links platziert.

Steht inverse auf true, so sind die Ergebnisse jeweils umgekehrt.

### 6.26.10 chart.visualMap.precision

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Die dezimale Präzision der Beschriftung, standardmäßig 0 (keine Dezimalstellen).

Im **CONTINUOUS-AVERAGE** Modus (wenn visualMap.splitNumber benutzt wird) ist die Regel des Layouts identisch zu visualMap.inverse, die Dezimalstellen passen sich automatisch an series.data an.

Im **CONTINUOUS-CUSTOMIZED** Modus (wenn visualMap.pieces benutzt wird) und im **CATEGORY** Modus (wenn visualMap.categories benutzt wird) ist die dezimale Präzision standardmäßig 0 (keine Dezimalstellen).

### 6.26.11 chart.visualMap.itemWidth

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Breite der Hauptleiste der stetigen visualMap-Komponente, oder die Breite jedes grafischen Elements, welches ein Stück in der stückweisen visualMap-Komponente repräsentiert.

### 6.26.12 chart.visualMap.itemHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Höhe der Hauptleiste der stetigen visualMap-Komponente, oder die Höhe jedes grafischen Elements, welches ein Stück in der stückweisen visualMap-Komponente repräsentiert.

### 6.26.13 chart.visualMap.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Spezifiziert die Positionierung von Handles und Beschriftungen an der Hauptleiste. Mögliche Werte sind:

- 'auto': Automatisch entscheiden.
- 'left': Die Handles und Beschriftungen stehen auf der rechten Seite. Gültig, wenn orient auf 'vertical' steht.
- 'right': Die Handles und Beschriftungen stehen auf der linken Seite. Gültig, wenn orient auf 'vertical' steht.
- 'top': Die Handles und Beschriftungen stehen auf der unteren Seite. Gültig, wenn orient auf 'horizontal' steht.
- 'bottom': Die Handles und Beschriftungen stehen auf der oberen Seite. Gültig, wenn orient auf 'horizontal' steht.



## Erlaubte Werte

- auto
- left
- right
- top
- bottom

### 6.26.14 chart.visualMap.text

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: null ]

Die Beschriftung an beiden Enden der Skala, zum Beispiel ['High', 'Low'].

Die Regel, dass Beschriftungen nicht angezeigt werden wenn text verwendet wird, wird für Kompatibilität mit ECharts2 erhalten.

### 6.26.15 chart.visualMap.textGap

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 10 ]

#### **Für stetige visualMap**

Die Distanz zwischen den Enden der Hauptleiste und den Beschriftungen. Einheit ist px. Siehe visualMap.text.

#### **Für stückweise visualMap**

Die Distanz zwischen den Enden der grafischen Elemente der Stücke und den Beschriftungen. Einheit ist px. Siehe visualMap.text.

### 6.26.16 chart.visualMap.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Spezifiziert, ob die visualMap-Komponente angezeigt werden soll. Steht der Wert auf false, wird die Komponente nicht angezeigt, kann aber immer noch visuelles Mapping von Datenwerten zu visuellen Kanälen im Diagramm durchführen.

### 6.26.17 chart.visualMap.dimension

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, welche Dimension verwendet werden soll, um Datenwerte aus series.data zu holen, die dann auf die visuellen Kanäle gemappt werden.

series.data kann als zweidimensionales Array betrachtet werden, zum Beispiel:

```
[
  [12, 23, 43],
  [12, 23, 43],
  [43, 545, 65],
  [92, 23, 33]
]
```

Jede Spalte des obigen Arrays wird als Dimension betrachtet. Wird visualMap.dimension zum Beispiel auf 1 gesetzt, so wird die zweite Spalte (23, 23, 545, 23) für das visuelle Mapping ausgewählt.

Verwendet standardmäßig die letzte Dimension der Daten.

### 6.26.18 chart.visualMap.seriesIndex

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, auf welchen Serien visuelles Mapping durchgeführt werden soll, von welchen Serien Daten geholt werden.

Standardmäßig werden alle Serien verwendet.

### 6.26.19 chart.visualMap.hoverLink

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

hoverLink aktiviert das Hervorheben bestimmter grafischer Elemente des Diagramms wenn die Maus auf einen Punkt der visualMap zeigt, der über visuelles Mapping mit diesen grafischen Elementen verbunden ist.

Im umgekehrten Fall wird, wenn die Maus auf ein grafisches Element des Diagramms zeigt, dessen Wertbeschriftung an seiner verbundenen Stelle in der visualMap angezeigt.

### 6.26.20 chart.visualMap.inRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

inRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    inRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      inRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    inRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in inRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  inRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    inRange: {color: ['red', 'blue']}
  }
});
```

**Hinweis:**

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.inRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.inRange.color = ['red', 'blue'];
option.visualMap.controller.inRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

**Hinweis:** Es gibt Standardfarben ['#f6efa6', '#d88273', '#bf444c'] in inRange wenn inRange nicht gesetzt wird. Sind diese unerwünscht, können sie durch inRange: { color: null } deaktiviert werden.

### 6.26.21 chart.visualMap.outOfRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

outOfRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    outOfRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      outOfRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    outOfRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in outOfRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  outOfRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    outOfRange: {color: ['red', 'blue']}
  }
});
```

## Hinweis:

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.outOfRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.outOfRange.color = ['red', 'blue'];
option.visualMap.controller.outOfRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

## 6.26.22 chart.visualMap.controller.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die Eigenschaften inRange und outOfRange können im controller-Objekt gesetzt werden, wodurch diese Werte für für inRange und outOfRange nur vom Controller (der visualMap-Komponente selbst) und nicht vom Diagramm (der Serie) benutzt werden. Diese Eigenschaft ist nützlich in Szenarien, in denen die Ansicht des Controllers im Detail angepasst werden muss.

### 6.26.23 chart.visualMap.controller.inRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

inRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen,...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    Komponente selbst:
    inRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [30, 100]
    }
  }
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      inRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    inRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      inRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in inRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.



Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  inRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    inRange: {color: ['red', 'blue']}
  }
});
```

**Hinweis:**

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.inRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.inRange.color = ['red', 'blue'];
option.visualMap.controller.inRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

**Hinweis:** Es gibt Standardfarben ['#f6efa6', '#d88273', '#bf444c'] in inRange wenn inRange nicht gesetzt wird. Sind diese unerwünscht, können sie durch inRange: { color: null } deaktiviert werden.

### 6.26.24 chart.visualMap.controller.outOfRange

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Definiere visuelle Kanäle, auf die von Datenwerten gemappt wird, die **im ausgewählten Bereich** liegen. (Nutzer können mit der visualMap-Komponente interagieren und einen Bereich mit der Maus oder Berührungen auswählen.)

Mögliche visuelle Kanäle sind:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL
- colorSaturation: Farbsättigung in HSL
- colorHue: Farbwert in HSL

outOfRange kann visuelle Kanäle sowohl in Serien (durch visualMap.seriesIndex) als auch in visualMap selbst anpassen.

Zum Beispiel, wenn eine visualMap-Komponente mit einem Streudiagramm verwendet wird, kann der Mapping-Ansatz von Daten zu Farben (oder Symbolen, Größen, ...) sowohl im Streudiagramm als auch in der visualMap-Komponente selbst angepasst werden. Siehe den folgenden Code:

```
visualMap: [  
  {  
    ...,  
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-  
    Komponente selbst:  
    outOfRange: {  
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],  
      symbolSize: [30, 100]  
    }  
  }  
]
```

Wenn die visuellen Kanäle für die Zielserie und die visualMap-Komponente getrennt definiert werden sollen, sollte der folgende Ansatz verwendet werden:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle nur für die Zielserie.
    target: {
      outOfRange: {
        color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
        symbolSize: [60, 200]
      }
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente.
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

Oder eine Definition wie folgt:

```

visualMap: [
  {
    ...,
    // Definiere visuelle Kanäle für sowohl die Zielserie als auch die visualMap-
    // Komponente selbst:
    outOfRange: {
      color: ['#121122', 'rgba(3,4,5,0.4)', 'red'],
      symbolSize: [60, 200]
    },
    // Definiere visuelle Kanäle nur für die visualMap-Komponente, welche
    // die Definitionen mit dem gleichen Namen von oben überschreiben
    // (symbolSize wird mit [30, 100] überschrieben während
    // color den vorherigen Wert behält)
    controller: {
      outOfRange: {
        symbolSize: [30, 100]
      }
    }
  }
]

```

## Über visuelle Kanäle

- Verschiedene visuelle Kanäle (wie color, symbolSize etc.) können in outOfRange auf einmal definiert werden und alle werden aufgenommen.
- Die Konfiguration von opacity statt colorAlpha wird empfohlen. Ersteres kontrolliert die Transparenz des grafischen Elements und seinen Anhängen (wie Beschriftungen), während letzteres nur die Transparenz des grafischen Elements kontrolliert
- Es werden zwei Ansätze zum visuellen Mapping unterstützt: 'Linear Mapping' und 'Table Mapping'.

## Lineares Mapping auf visuelle Kanäle

Lineares Mapping bedeutet, dass auf jedem Datenwert (Wert aus `series.data`) eine lineare Berechnung durchgeführt wird, wodurch sie aus dem Bereich von `[visualMap.min, visualMap.max]` in den Bereich von [visueller Wert 1, visueller Wert 2] gemappt werden und ein finaler Wert (z.B. ein visueller Wert) für das Rendern des visuellen Kanals erhalten wird.

Zum Beispiel, sei `[visualMap.min, visualMap.max]` auf `[0, 100]` gesetzt, und sei `series.data` `[50, 10, 100]`. Diese Werte sollen auf den `opacity`-Bereich `[0.4, 1]` gemappt werden, durch den die Größe eines Wertes durch die Undurchsichtigkeit seines grafischen Elements dargestellt wird. Die `visualMap`-Komponente berechnet dann linear die `opacity`-Werte `[0.7, 0.44, 1]` für die jeweiligen Datenwerte.

Der visuelle Bereich kann auch umgekehrt gesetzt werden, wie zum Beispiel `opacity: [1, 0.4]`, wodurch das Endergebnis für die obige Serie `[0.7, 0.96, 0.4]` beträgt.

Hinweis: `[visualMap.min, visualMap.max]` sollte manuell gesetzt werden und ist standardmäßig `[0, 200]`, nicht `dataMin` und `dataMax` aus `series.data`.

Wie muss die `visualMap`-Komponente für lineares Mapping konfiguriert sein?

- Eine stetige `visualMap` wird verwendet, oder
- Eine stückweise `visualMap` wird verwendet und `visualMap.categories` ist nicht gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

- Grundlegend wird ein Array verwendet um den Bereich des visuellen Wertes darzustellen, z.B. `color: ['#333', '#777']`.
- Einzelne Zahlen oder Strings können auch verwendet werden, die `visualMap`-Komponente wandelt solche in ein Array um. Zum Beispiel: `opacity: 0.4` wird konvertiert zu `opacity: [0.4, 0.4]`, `color: '#333'` wird konvertiert zu `color: ['#333', '#333']`.
- Für die visuellen Kanäle `symbolSize`, `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation` und `colorHue` hat der Bereich des visuellen Wertes immer die Form: [visueller Wert von `visualMap.min`, visueller Wert von `visualMap.max`]. Zum Beispiel: `colorLightness: [0.8, 0.2]` bedeutet, dass ein Datenwert in `series.data` der gleich `visualMap.min` ist, falls er existiert, auf eine Helligkeit von 0.8 gemappt wird. Weiterhin wird ein Datenwert der gleich `visualMap.max` ist, falls er existiert, auf eine Helligkeit von 0.2 gemappt. Andere Datenwerte werden anhand einer linearen Berechnung basierend auf den Bereichen `[visualMap.min, visualMap.max]` und `[0.8, 0.2]` gemappt.
- Für den visuellen Kanal `color` wird ein Array wie `['#333', '#78ab23', 'blue']` verwendet, durch das ein Farbband anhand der drei Farbpunkte definiert wird. Datenwerte werden dann auf dieses Farbband gemappt. Insbesondere werden Datenwerte, die gleich `visualMap.min` sind, auf `'#333'` gemappt, Datenwerte, die gleich `visualMap.max` sind, auf `'blue'` gemappt, und andere Datenwerte stückweise interpoliert, um die endgültige Farbe zu erhalten.
- Für den visuellen Kanal `symbol` wird ein Array wie `['circle', 'rect', 'diamond']` verwendet, mit dem Datenwerte, die gleich `visualMap.min` sind, auf `'circle'` gemappt werden, Datenwerte, die gleich `visualMap.max` sind, auf `'diamond'` gemappt werden, und andere Datenwerte anhand ihrer numerischen Distanz zu `visualMap.min` und `visualMap.max` auf `'circle'`, `'rect'` oder `'diamond'` gemappt werden.

Über die möglichen Wertebereiche von visuellen Werten:

- `opacity`, `colorAlpha`, `colorLightness`, `colorSaturation`  
Die möglichen Werte liegen in `[0, 1]`.
- `colorHue`  
Die möglichen Werte liegen in `[0, 360]`.
- `color`  
Mögliche Werte sind RGB-Ausdrücke, wie `'rgb(128, 128, 128)'`, RGBA-Ausdrücke, wie `'rgba(128, 128, 128, 0.5)'`, oder Hex-Ausdrücke, wie `'#ccc'`.
- `symbol`  
Von ECharts bereitgestellte Icon-Typen sind `'circle'`, `'rect'`, `'roundRect'`, `'triangle'`, `'diamond'`, `'pin'`, `'arrow'` und `'none'`.  
Kann auf ein Bild gesetzt werden über `'image://url'`, wobei URL der Link zu einem Bild ist, oder über einen `dataURI` eines Bildes.

Icons können in ECharts über 'path:/' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### Table Mapping auf visuelle Kanäle

Table Mapping kann verwendet werden wenn die Datenwerte (Werte in series.data, spezifiziert durch visualMap.dimension) abzählbar sind und wir sie anhand einer gegebenen Tabelle auf visuelle Werte mappen wollen.

Zum Beispiel, sei in einer stückweisen visualMap-Komponente visualMap.categories auf ['Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Sei außerdem series.data auf ['Demon Hunter', 'Death Knight', 'Warden', 'Paladin'] gesetzt. Dann können wir die folgende lookup-Regel für color erstellen: color: {'Warden': 'red', 'Demon Hunter': 'black'}, wodurch die visualMap-Komponente Datenwerte auf Farben mappt.

Wie muss die visualMap-Komponente für Table Mapping konfiguriert sein?

- Eine stückweise visualMap wird verwendet und visualMap.categories ist gesetzt.

Über den Wert des visuellen Kanals (visueller Wert):

Im Allgemeinen werden Objekte oder Arrays verwendet, zum Beispiel:

```
visualMap: {
  type: 'piecwise',
  // categories definiert die Elemente, die in der visualMap-Komponente dargestellt
  // werden sollen.
  categories: [
    'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
  ],
  outOfRange: {
    // Der visuelle Wert kann ein Objekt sein?
    color: {
      'Warden': 'red',
      'Demon Hunter': 'black',
      '': 'green' // Ein leerer String bedeutet, dass außer 'Warden' und 'Demon
Hunter'
                // alle Datenwerte auf 'green' gemappt werden sollen.
    }
    // Der visuelle Wert kann auch ein einzelner Wert sein,
    // wodurch alle Datenwerte auf diesen Wert gemappt werden.
    color: 'green',
    // Der visuelle Wert kann auch ein Array sein, mit der gleichen
    // Länge wie das Array der Kategorien und einem eins-zu-eins mapping
    color: ['red', 'black', 'green', 'yellow', 'white']
  }
}
```

### Wie modifiziert man die Konfiguration des visuellen Encodings?

Soll die Konfiguration des visuellen Encodings verändert werden, nachdem das Diagramm gerendert wurde (indem setOption aufgerufen wurde um die initiale Option zu setzen), kann setOption erneut verwendet werden, um die Konfiguration des visuellen Encodings zu verändern. Zum Beispiel:

```
chart.setOption({
  visualMap: {
    outOfRange: {color: ['red', 'blue']}
  }
});
```

## Hinweis:

- Diese visualMap-Eigenschaften (inRange, outOfRange, target, controller) unterstützen kein "merge", das heißt: Alle davon werden modifiziert, wenn setOption erneut aufgerufen wird, und alle vorherigen Werte werden gelöscht. Der "merge" verkompliziert die Implementation und das Verständnis, wohingegen "alles löschen" den Vorgang normalisiert: Sollen einzelne visuelle Werte verändert werden, sollten alle visuellen Werte an setOption übergeben werden, egal ob sie verändert wurden, oder nicht.
- Daher wird getOption() -> modifiziere die erhaltene Option -> setOption(modifizierte Option), strengstens **nicht empfohlen**, zum Beispiel:

```
// Nicht empfohlener Ansatz, ungeachtet seiner Korrektheit:

var option = chart.getOption(); // Gesamte Option holen.
option.visualMap.outOfRange.color = ['red', 'blue']; // Farbe modifizieren, was das
gewollte Ergebnis ist.

// Diese beiden Eigenschaften müssen modifiziert werden, sonst wird nicht das
gewünschte Ergebnis erreicht.
option.visualMap.target.outOfRange.color = ['red', 'blue'];
option.visualMap.controller.outOfRange.color = ['red', 'blue'];

chart.setOption(option); // Modifizierte Option setzen.
// Dieser Ansatz sollte nicht verwendet werden, sondern der
// Ansatz, der vor diesem Beispiel demonstriert wurde.
```

## 6.26.25 chart.visualMap.zlevel

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

zlevel-Wert aller grafischen Elemente im Container.

zlevel wird benutzt, um mit Canvas-Elementen Ebenen zu erstellen. Grafische Elemente mit verschiedenen zlevel-Werten werden auf verschiedene Canvas-Elemente platziert, was eine geläufige Optimierungstechnik ist. Wir können Elemente, die häufig verändert werden, (wie solche mit Animationen) auf ein separates zlevel legen. Es ist zu bedenken, dass zu viele Canvas-Elemente die Speicherkosten erhöhen, und auf Mobilgeräten vorsichtig eingesetzt werden sollten, um Abstürze zu vermeiden.

Canvas-Elemente mit höherem zlevel werden über Canvas-Elementen mit niedrigerem zlevel platziert.

### 6.26.26 chart.visualMap.z

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 4 ]

z-Wert aller grafischen Elemente im Container, welcher die Reihenfolge des Zeichnens grafischer Komponenten kontrolliert. Komponenten mit kleineren z-Werten können von solchen mit größeren z-Werten überschrieben werden.

z hat eine niedrigere Priorität als zlevel und erstellt kein neues Canvas-Element.

### 6.26.27 chart.visualMap.left

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Abstand zwischen der visualMap-Komponente und der linken Seite des Containers.

Der Wert von left kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Breite des Containers wie '20%' sein; und er kann auch 'left', 'center' oder 'right' sein.

Wenn der Wert von left auf 'left', 'center' oder 'right' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.



## Erlaubte Werte

- {percent}
- {integer}
- left
- center
- right

### 6.26.28 chart.visualMap.top

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'auto' ]

Abstand zwischen der visualMap-Komponente und der oberen Seite des Containers.

Der Wert von top kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein; und er kann auch 'top', 'middle' oder 'bottom' sein.

Wenn der Wert von top auf 'top', 'middle' oder 'bottom' gesetzt ist, wird die Komponente automatisch anhand der Position ausgerichtet.

## Erlaubte Werte

- {percent}
- {integer}
- top
- middle
- bottom
- auto

### 6.26.29 chart.visualMap.right

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: 'auto' ]

Abstand zwischen der visualMap-Komponente und der rechten Seite des Containers

Der Wert von right kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zu der Breite des Containers wie '20%' sein.

### Erlaubte Werte

- {percent}
- {integer}
- auto

## 6.26.30 chart.visualMap.bottom

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Abstand zwischen der visualMap-Komponente und der unteren Seite des Containers.

Der Wert von bottom kann ein direkter Pixelwert wie 20 sein; er kann auch ein prozentualer Wert relativ zur Höhe des Containers wie '20%' sein.

### Erlaubte Werte

- {integer}
- {percent}

## 6.26.31 chart.visualMap.orient

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'vertical' ]

Wie das Layout der visualMap-Komponente aussehen soll.

#### Erlaubte Werte

- horizontal
- vertical

### 6.26.32 chart.visualMap.padding

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Innenabstand in der visualMap um den Inhalt. Die Einheit ist px. Der standardmäßige Wert für jede Position ist 5. Sie können auf andere Werte gesetzt werden mit left, right, top und bottom.

Beispiele:

```
// Setze Innenabstand auf 5
padding: 5
// Setze den oberen und unteren Innenabstand auf 5, sowie den linken und rechten
Innenabstand auf 10
padding: [5, 10]
// Setze jeden der vier Innenabstände separat
padding: [
  5, // oben
  10, // rechts
  5, // unten
  10, // links
]
```

### 6.26.33 chart.visualMap.backgroundColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'rgba(0,0,0,0)' ]

Hintergrundfarbe der visualMap-Komponente

### 6.26.34 chart.visualMap.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#ccc' ]

Rahmenfarbe der visualMap-Komponente.

### 6.26.35 chart.visualMap.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite der visualMap-Komponente, mit Einheit px.

### 6.26.36 chart.visualMap.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: ['#bf444c', '#d88273', '#f6efa6'] ]

Diese Eigenschaft verbleibt nur für Kompatibilität mit ECharts 2 und ihre Verwendung in ECharts 3 wird nicht empfohlen. Farben sollten stattdessen in `visualMap.inRange` oder `visualMap.outOfRange` konfiguriert werden.

Wird die Eigenschaft weiter verwendet, sollte das folgende Problem bemerkt werden: Die Reihenfolge, in der die Datenwerte auf Farbwerte aus der Eigenschaft `color` gemappt werden, ist von groß nach klein, während sie für die Eigenschaften `visualMap.inRange` und `visualMap.outOfRange` von klein nach groß ist.

### 6.26.37 chart.visualMap.textStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Stil des Textes der `visualMap`.

### 6.26.38 chart.visualMap.textStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

### Schlagworte / Tags

[ Default-Wert: #333 ]

Textfarbe.

## 6.26.39 chart.visualMap.textStyle.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'normal' ]

Schriftstil.

### Erlaubte Werte

- normal
- italic
- oblique

## 6.26.40 chart.visualMap.textStyle.fontWeight

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ default: normal ]

Schriftstärke

Mögliche Werte:

- 'normal'
- 'bold'
- 'bolder'

- 'lighter'
- 100 | 200 | 300 | 400...

## Erlaubte Werte

- normal
- bold
- bolder
- lighter
- {integer}

### 6.26.41 chart.visualMap.textStyle.fontFamily

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'sans-serif' ]

visualMap Schriftfamilie.

Akzeptiert auch 'serif', 'monospace', ...

### 6.26.42 chart.visualMap.textStyle.fontSize

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 12 ]

Schriftgröße.

### 6.26.43 chart.visualMap.textStyle.lineHeight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Zeilenhöhe des Textfragments

Wenn lineHeight im rich nicht gesetzt wird, wird die lineHeight der nächsthöheren Ebene verwendet. Zum Beispiel:

```
{
  lineHeight: 56,
  rich: {
    a: {
      // `lineHeight` ist nicht gesetzt, wird zu 56
    }
  }
}
```

### 6.26.44 chart.visualMap.textStyle.width

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Breite des Textblocks. Ist standardmäßig die Breite des Texts. In den meisten Fällen ist es nicht nötig diesen Wert zu spezifizieren. Er könnte in Fällen wie bei der Erstellung einfacher Tabellen oder der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

width kann auch ein prozentualer String sein, wie '100%', welcher sich auf den Prozentanteil von contentWidth (also der Breite ohne Innenabstand) seiner Container Box bezieht. Es basiert auf contentWidth da jedes



Textfragment in seinem Layout von der content Box abhängt, wo es in der Praxis keinen Sinn macht, die Breite anhand der äußeren Breite zu berechnen.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.26.45 chart.visualMap.textStyle.height

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Höhe des Textblocks. Ist standardmäßig die Höhe des Textes. Könnte in Fällen wie bei der Benutzung von Hintergrundbildern Verwendung finden (siehe backgroundColor).

Es ist anzumerken, dass width und height die Breite und Höhe des Inhalts spezifizieren, ohne Innenabstand.

Es ist anzumerken, dass width und height nur funktionieren, wenn rich spezifiziert ist.

### 6.26.46 chart.visualMap.textStyle.textBorderColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Rahmenfarbe des Textes selbst.

### 6.26.47 chart.visualMap.textStyle.textBorderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite des Textes selbst.

### 6.26.48 chart.visualMap.textStyle.textShadowColor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'transparent' ]

Schattenfarbe des Textes selbst.

### 6.26.49 chart.visualMap.textStyle.textShadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Schattenverwischung des Textes selbst.

### 6.26.50 chart.visualMap.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
-----------------	---------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der X-Achse.

### 6.26.51 chart.visualMap.textStyle.textShadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Verschiebung des Schattens vom Text selbst entlang der Y-Achse.

### 6.26.52 chart.visualMap.formatter

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Der Formatierer für die Beschriftung.

- Wird er als String gesetzt, steht er für eine String-Template, zum Beispiel : 'aaaa{value}bbbb', wobei {value} den Wert am Rand des ausgewählten Bereichs darstellt.
- Wird er als Funktion gesetzt, steht er für eine Callback-Funktion, zum Beispiel:

```
formatter: function (value) {
  return 'aaaa' + value + 'bbbb';
}
```

### 6.26.53 chart.visualMap.splitNumber

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 5 ]

Stetige Daten können nach splitNumber in Stücke unterteilt werden. Das heißt, wenn splitNumber 5 ist, werden die Daten in 5 Stücke unterteilt.

Der Bereich der stetigen Daten sollte mit max und min definiert werden.

Wenn visualMap.pieces oder visualMap.categories gesetzt ist, wird splitNumber nicht mehr verwendet.

### 6.26.54 chart.visualMap.pieces

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wird verwendet um anzupassen, wie stetige Daten aufgeteilt werden sollen, und um spezifische Ansichten für solche Stücke zu definieren. Zum Beispiel:

```
pieces: [
  // Der Bereich eines Stücks kann über die Eigenschaften min und max definiert
  werden,
  // wobei min auf -Infinity gesetzt wird, falls ignoriert,
  // und max auf Infinity gesetzt wird, falls ignoriert.
  {min: 1500},
  {min: 900, max: 1500},
  {min: 310, max: 1000},
  {min: 200, max: 300},
  // Die Beschriftung eines Stücks kann spezifiziert werden.
  {min: 10, max: 200, label: '10 to 200 (custom label) '},
  // Die Farbe eines Stücks kann spezifiziert werden.
  {value: 123, label: '123 (custom special color) ', color: 'grey'},
  {max: 5}
]
```

Die folgenden visuellen Kanäle können für jedes Stück angepasst werden:

- symbol: Symboltyp.
- symbolSize: Symbolgröße.
- color: Symbolfarbe.
- colorAlpha: Alphakanal des Symbols.
- opacity: Undurchsichtigkeit des Symbols und anderem (wie Beschriftungen).
- colorLightness: Helligkeit in HSL.
- colorSaturation: Farbsättigung in HSL.
- colorHue: Farbton in HSL.

Hinweis: In ECharts2 heißt pieces splitList, was in ECharts3 für Kompatibilität erhalten bleibt. Es wird aber empfohlen, pieces zu verwenden.

Die Sequenz der Stücke lässt sich über einfaches ausprobieren herausfinden. Für weitere Informationen siehe visualMap.inverse.

### 6.26.55 chart.visualMap.categories

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Wenn die Datenwerte in series.data (spezifiziert durch visualMap.dimension) diskret sind (auch bekannt als Kategoriedaten oder abzählbare Daten), und ein **Table Mapping** von den Datenwerten zu visuellen Kanälen vorgenommen werden soll, wird categories verwendet, um die gesamte Aufzählung der Daten zu beschreiben. Zum Beispiel:

```
categories: [
  'Demon Hunter', 'Blademaster', 'Death Knight', 'Warden', 'Paladin'
],
```

Die Reihenfolge in categories kann durch ausprobieren erkannt werden. Für weitere Informationen siehe `visualMap.inverse`.

### 6.26.56 chart.visualMap.minOpen

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option funktioniert wenn type auf 'piecewise' steht und min/max/splitNumber gesetzt sind. Steht dieser Wert auf true, so wird ein zusätzliches Stück mit der Beschriftung "< min" angezeigt.

### 6.26.57 chart.visualMap.maxOpen

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Diese Option funktioniert wenn type auf 'piecewise' steht und min/max/splitNumber gesetzt sind. Steht dieser Wert auf true, so wird ein zusätzliches Stück mit der Beschriftung "> max" angezeigt.

### 6.26.58 chart.visualMap.selectedMode

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'multiple' ]

Wechselt zwischen Einzelauswahl und multipler Auswahl.

Erlaubte Werte

- single
- multiple

### 6.26.59 chart.visualMap.showLabel

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Spezifiziert, ob die Beschriftung jedes Elements angezeigt werden soll. Standardmäßig wird die Beschriftung nicht angezeigt, wenn visualMap.text verwendet wird, sonst wird sie angezeigt.

### 6.26.60 chart.visualMap.textStyle.textBorderStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Strichlinientyp des Textes. Mögliche Werte sind:

- 'solid'
- 'dashed'
- 'dotted'

Seit v5.0.0 kann es auch eine Zahl oder ein Zahlen-Array sein, um das Bindestrich-Array der Zeile anzugeben. Mit `textBorderDashOffset` können wir den Linienstil flexibler gestalten.

Zum Beispiel:

```
{ textBorderType: [5, 10], textBorderDashOffset: 5 }
```

Erlaubte Werte

- dashed
- dotted
- solid

### 6.26.61 chart.visualMap.textStyle.textBorderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `textBorderType` können wir den Linienstil flexibler gestalten.

### 6.26.62 chart.visualMap.textStyle.overflow

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Bestimmt, wie der Text angezeigt werden soll, wenn er überläuft. Verfügbar, wenn die Breite eingestellt ist.

- 'truncate' Den Text abschneiden und mit Auslassungspunkten nachstellen.
- 'break' Wortweise brechen



- 'breakAll' Umbruch nach Zeichen.

## Erlaubte Werte

- truncate
- break
- breakAll

### 6.26.63 chart.visualMap.textStyle.ellipsis

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Auslassungspunkte, die angezeigt werden sollen, wenn der Überlauf auf Abschneiden eingestellt ist.

- 'truncate' Kürzt die Überlaufzeilen.

### 6.26.64 chart.visualMap.handleIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'M8.2,13.6V3.9H6.3v9.7H3.1v14.9h3.3v9.7h1.8v-9.7h3.3V13.6H8.2z M9.7,24.4H4.8v-1.4h4.9V24.4z M9.7,19.1H4.8v-1.4h4.9V19.1z' ]

Bildliche Form des Handles, welche Pfadstrings unterstützt.

Kann über 'image://url' auf ein Bild gesetzt werden, wobei URL der Link zu einem Bild ist, oder über einen dataURI eines Bildes.

Icons können in ECharts über 'path://' auf beliebige Vektorpfade gesetzt werden. Verglichen mit Rasterbildern verhindern Vektorpfade die Zackenbildung und Verwischung beim Skalieren und haben mehr Kontrolle über Farbveränderung. Die Größe des vektorisierten Icons wird automatisch angepasst. Für mehr Informationen über

das Format von Pfaden beziehe man sich auf SVG PathData. Vektorpfade lassen sich aus Tools wie Adobe Illustrator exportieren.

### 6.26.65 chart.visualMap.handleSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '120%' ]

Größe des Ziehgriffs. Es kann eine Prozentzeichenfolge sein.

Erlaubte Werte

- {integer}
- {percent}

### 6.26.66 chart.visualMap.handleStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Art des Ziehgriffs.

### 6.26.67 chart.visualMap.handleStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: #a7b7cc ]

Farbe.

### 6.26.68 chart.visualMap.handleStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.26.69 chart.visualMap.handleStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.26.70 chart.visualMap.handleStyle.borderType

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

Erlaubte Werte

- solid
- dashed
- dotted

### 6.26.71 chart.visualMap.handleStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit borderType können wir den Linienstil flexibler gestalten.

### 6.26.72 chart.visualMap.handleStyle.borderCap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : butt

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- 'butt': Die Linienenden werden an den Endpunkten abgerundet.
- 'round': Die Linienenden werden abgerundet.
- 'square': Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- butt
- round
- square

### 6.26.73 chart.visualMap.handleStyle.borderJoin

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

### 6.26.74 chart.visualMap.handleStyle.borderMiterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.26.75 chart.visualMap.handleStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.26.76 chart.visualMap.handleStyle.shadowColor

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.26.77 chart.visualMap.handleStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.26.78 chart.visualMap.handleStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.26.79 chart.visualMap.handleStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

### 6.26.80 chart.visualMap.indicatorIcon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Symbol des Indikators.

### 6.26.81 chart.visualMap.indicatorSize

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '50%' ]

Größe des indicator. Es kann eine Prozentzeichenfolge sein.



## Erlaubte Werte

- {integer}
- {percent}

## 6.26.82 chart.visualMap.indicatorStyle.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Stil des Indikators.

## 6.26.83 chart.visualMap.indicatorStyle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: #a7b7cc ]

Farbe.

## 6.26.84 chart.visualMap.indicatorStyle.borderColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: '#000' ]

Rahmenfarbe, deren Format ähnlich zu dem von color ist.

### 6.26.85 chart.visualMap.indicatorStyle.borderWidth

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Rahmenbreite. Wenn dieser Wert auf 0 steht, wird kein Rahmen angezeigt.

### 6.26.86 chart.visualMap.indicatorStyle.borderType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 'solid' ]

Rahmentyp.

#### Erlaubte Werte

- solid
- dashed
- dotted

### 6.26.87 chart.visualMap.indicatorStyle.borderDashOffset

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Um den Linienstrich-Offset einzustellen. Mit `borderType` können wir den Linienstil flexibler gestalten.

### 6.26.88 chart.visualMap.indicatorStyle.borderCap

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : `butt`

Legt fest, wie die Endpunkte der Linie gezeichnet werden.

Mögliche Werte sind:

- `'butt'`: Die Linienenden werden an den Endpunkten abgerundet.
- `'round'`: Die Linienenden werden abgerundet.
- `'square'`: Die Enden von Linien werden durch Hinzufügen eines Kästchens mit gleicher Breite und halber Höhe der Linienstärke abgerundet.

Erlaubte Werte

- `butt`
- `round`
- `square`

### 6.26.89 chart.visualMap.indicatorStyle.borderJoin

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : bevel

Um die Form zu bestimmen, die verwendet wird, um zwei Liniensegmente dort zu verbinden, wo sie sich treffen.

Mögliche Werte sind:

- 'bevel': Füllt einen zusätzlichen dreieckigen Bereich zwischen dem gemeinsamen Endpunkt verbundener Segmente und den separaten äußeren rechteckigen Ecken jedes Segments.
- 'round': Rundet die Ecken einer Form ab, indem ein zusätzlicher Scheibensektor ausgefüllt wird, der am gemeinsamen Endpunkt der verbundenen Segmente zentriert ist. Der Radius für diese abgerundeten Ecken ist gleich der Linienbreite.
- „miter“: Verbundene Segmente werden verbunden, indem ihre Außenkanten verlängert werden, um sie an einem einzigen Punkt zu verbinden, wodurch ein zusätzlicher rautenförmiger Bereich gefüllt wird. Diese Einstellung wird von der Eigenschaft miterLimit beeinflusst.

Erlaubte Werte

- bevel
- round
- miter

## 6.26.90 chart.visualMap.indicatorStyle.borderMiterLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Default : 10

Zum Einstellen des Gehrungsbegrenzungsverhältnisses. Funktioniert nur, wenn Verbinden als Gehrung eingestellt ist.

Der Standardwert ist 10. Negative,0,Infinity- und NaN-Werte werden ignoriert.

### 6.26.91 chart.visualMap.indicatorStyle.shadowBlur

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Größe der Schattenverwischung. Dieses Attribut sollte zusammen mit shadowColor, shadowOffsetX und shadowOffsetY verwendet werden, um den Schatten der Komponente zu setzen.

Zum Beispiel:

```
{
  shadowColor: 'rgba(0, 0, 0, 0.5)',
  shadowBlur: 10
}
```

### 6.26.92 chart.visualMap.indicatorStyle.shadowColor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Schattenfarbe. Unterstützt das gleiche Format wie color.

### 6.26.93 chart.visualMap.indicatorStyle.shadowOffsetX

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in horizontale Richtung.

### 6.26.94 chart.visualMap.indicatorStyle.shadowOffsetY

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

[ Default-Wert: 0 ]

Offset-Distanz des Schattens in vertikale Richtung.

### 6.26.95 chart.visualMap.indicatorStyle.opacity

<b>Datentyp</b>	DECIMAL
<b>Semantische Gruppe</b>	Visual Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	5.0.0
<b>Schlagworte / Tags</b>	

Undurchsichtigkeit der Komponente. Unterstützt Werte von 0 bis 1 und die Komponente wird nicht gezeichnet, wenn auf 0 gestellt.

## 7 Leaflet [Options]

<b>Name</b>	Leaflet
<b>Version</b>	1.6
<b>Hersteller</b>	© 2010–2019 Vladimir Agafonkin. Maps © OpenStreetMap contributors.
<b>Webseite</b>	<a href="https://leafletjs.com/">https://leafletjs.com/</a>

### 7.1 Leaflet Buttons

#### 7.1.1 leaflet.distancebutton.values

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Buttons
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: 10;25;50

Option zum Rendern der Buttons, die die Werte (in Kilometer) über dem Leaflet-Dashlet anzeigen würden. Die Leaflet-Marker / Geokoordinaten innerhalb von Kilometerwerten werden dann gefiltert, wenn auf den Button geklickt wird.

Die Liste der Kilometerwerte sollte durch Semikolon getrennt sein.

Beispiel:

```
dashlet.addOption("magic.add.buttonbar", "true");
dashlet.addOption("leaflet.distancebutton.values", "2;8;10;15"); // in km
```

### 7.2 Leaflet General

#### 7.2.1 leaflet.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: 800

Legt die Höhe des Leaflet-Dashlets fest.

```
dashlet.addOption("leaflet.height", "600");
```

## 7.3 Leaflet Infobox

### 7.3.1 leaflet.infobox.style

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Infobox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: ""

Legt den Stil der Infobox fest.

```
dashlet.addOption("leaflet.infobox.style", "background-color: white; padding: 10px; text-align: center;");
```



### 7.3.2 leaflet.infobox.style.default

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Infobox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: ""

Legt den Standardstil der Infobox fest.

```
dashlet.addOption("leaflet.infobox.style.default", "background-color: yellow; padding: 10px; width: 500px; text-align: center;");
```

### 7.3.3 leaflet.infobox.html.active

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	Infobox
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: 'Active'

Legt den Inhalt fest, der im Infofeld angezeigt werden soll, wenn der Marker ausgewählt ist.

```
dashlet.addOption("leaflet.infobox.html.active", "'<b>' + data[\"Sparte\"] + '</b><br />' + data[\"Postleitzahl\"] + ' ' + data[\"Stadt\"] + '<br/>' + data[\"Anschrift\"]" );
```

### 7.3.4 leaflet.infobox.html.default

<b>Datentyp</b>	HTML
<b>Semantische Gruppe</b>	Infobox

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: 'Default'

Legt den Inhalt fest, der im Infobereich angezeigt werden soll, wenn der Marker nicht ausgewählt ist.

```
dashlet.addOption("leaflet.infobox.html.default", "");
```

## 7.4 Leaflet Layer

### 7.4.1 leaflet.layer.circle.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: false

Option zum Anzeigen einer Kreis-Ebene auf der Leaflet.

Beispiel:

```
dashlet.addOption("leaflet.layer.circle.show", "true");
```

### 7.4.2 leaflet.layer.circle.radius

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: 50000

Option zum Festlegen des Radius einer Kreis-Ebene auf der Leaflet.

Der Wert wird in Metern eingegeben.

Beispiel:

```
dashlet.addOption("leaflet.layer.circle.show", "true");
dashlet.addOption("leaflet.layer.circle.radius", "15000"); // in meters
```

### 7.4.3 leaflet.layer.circle.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: blue

Option zum Festlegen des Farbes einer Kreis-Ebene auf der Leaflet.

Beispiel:

```
<span style="font-size:12px">dashlet.addOption("leaflet.layer.circle.show", "true");
dashlet.addOption("leaflet.layer.circle.color", "red"); </span>
```

### 7.4.4 leaflet.layer.circle.fillcolor

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: #00f

Option zum Festlegen des Farbes für füllen einer Kreis-Ebene auf der Leaflet.

Beispiel:

```
dashlet.addOption("leaflet.layer.circle.show", "true");
dashlet.addOption("leaflet.layer.circle.fillcolor", "#0f0");
```

### 7.4.5 leaflet.layer.circle.opacity

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: 0.1

Option zum Festlegen des Opacity einer Kreis-Ebene auf der Leaflet.

Beispiel:

```
<span style="font-size:12px">dashlet.addOption("leaflet.layer.circle.show", "true");
dashlet.addOption("leaflet.layer.circle.</span><span style="font-size:11.0pt">.<span
style="font-size:12px">opacity", "0.5"); </span></span><span style="font-size:12px">
</span>
```

### 7.4.6 leaflet.layer.circle.weight

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Layer
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.10
<b>Schlagworte / Tags</b>	

Default Wert: 1

Option zum Festlegen des weicht einer Kreis-Ebene auf der Leaflet.

Beispiel:

```
<span style="font-size:12px">dashlet.addOption("leaflet.layer.circle.show", "true");
dashlet.addOption("leaflet.layer.circle.weight", "5");
</span>
```

## 7.5 Leaflet Map

### 7.5.1 leaflet.map.filename

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Ermöglicht das Hinzufügen einer Kartenebene (Map Layer) als Hintergrund für das Leaflet-Dashlet.

Der Pfad zur JSON-Map-Datei sollte angegeben werden.

```
dashlet.addOption("leaflet.map.filename", "/dashlets-showcase/getjson/maps/
bundeslaender.json");
```

### 7.5.2 leaflet.map.autoload

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: true

Wenn true, wird der Map-Layer als Hintergrund für das Leaflet Dashlet angezeigt, wenn das Dashboard ausgeführt wird.

Wird zusammen mit der Option "leaflet.map.filename" verwendet.

```
dashlet.addOption("leaflet.map.autoload", "false");
```

### 7.5.3 leaflet.map.tileLayer

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: `https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png`

Dient zum Laden und Anzeigen von Tile-Ebenen auf der Karte.

```
dashlet.addOption("leaflet.map.tileLayer", "http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png");
```

### 7.5.4 leaflet.map.viewpoint

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Map
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: [50.10208,9.06372]

Legt die Ansicht der Karte auf das angegebene geografische Zentrum fest

```
dashlet.addOption("leaflet.map.viewpoint", "[50.101694,8.631921]");
```

## 7.6 Leaflet Marker

### 7.6.1 leaflet.marker.options.title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Text für den Browser-Tooltip, der beim Marker-Hover angezeigt wird (standardmäßig kein Tooltip).

Hinweis: '\\n' wird verwendet, um den Text in der nächsten Zeile anzuzeigen.

```
dashlet.addOption("leaflet.marker.options.title", "mo['Postleitzahl'] + ' ' + mo['Stadt'] + '\\n' + mo['Anschrift']");
```

**Hinweis:** Auf dashBIRD-Studio bitte die Option für NewLine als '\\n' nutzen

### 7.6.2 leaflet.marker.event.mouseover

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktion für den Browser-Tooltip, der beim Mouseover-Event angezeigt wird.

```
dashlet.addOption("leaflet.marker.event.mouseover",
"leaflet1_info.update(param.target.options.data)");
```

### 7.6.3 leaflet.marker.event.mouseout

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Funktion für den Browser-Tooltip, der beim Mouseout-Event angezeigt wird.

```
dashlet.addOption("leaflet.marker.event.mouseout", "leaflet1_info.update(null)");
```

### 7.6.4 leaflet.marker.icon.active

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Marker auf das entsprechende Icon, wenn er ausgewählt ist.

Hinweis: Die Details des Icons müssen mithilfe der Icon-Optionen separat angegeben werden.

```
dashlet.addOption("leaflet.marker.icon.active","leaflet1_icon2");
```

### 7.6.5 leaflet.marker.icon.default

<b>Datentyp</b>	STRING
-----------------	--------



<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Marker auf das entsprechende Icon, wenn er nicht ausgewählt ist (Inaktiv).

Hinweis: Die Details des Icons müssen mithilfe der Icon-Optionen separat angegeben werden.

```
dashlet.addOption("leaflet.marker.icon.default","leaflet1_icon1");
```

### 7.6.6 leaflet.marker.icon.[].iconUrl

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Die URL zum Iconbild (absolut oder relativ zum Skriptpfad).

Beispiel: Das Icon ist hier als Base64 Bild gesetzt.

```
dashlet.addOption(leaflet.marker.icon.1.iconUrl", "data:image/
png;base64,iVBORw0KGgoAAAANSUHEUgAAAOEAAADhCAMAAAAJbSJIAAAAkFBMVEX/////AAD//Pz/+Pj/
9PT/+vr/QUH/5+f/3d3/7u7/xcX/fX3/y8v/MDD/np7/6+v/c3P/U1P/b2//Njb/4uL/jY3/wMD/k5P/0tL/
YWH/0zv/eHj/RUX/sLD/19f/FBT/pKT/urr/Jyf/Hx//Wlr/Kyv/Zmb/DQ3/g4P/ra3/mpr/TU3/VVX/iIj/
qan/
RERLWDTfAAAj6kleQVR4n02diXqiQAYAW0BFxRvxREXR3eP9325rt922HklmJgm03/4PwBADJnfm7k4er+T7Q
VzbN/v1dwbNNKoEvL/yfJaXJWjFs3T3sL6/Smp0lM3ilp/3W9rih1Hy1Jlff+6T1cMgieJS3m9rTBCL/
UdUui9SptFPEtKLDu0xVboPyvXmM08XJ9JqbozFe2Nb3iyCvN8eZ9htbK3ke2PeaMd5SwDiZQ176T6Y1gqrQ1
p7Bvl0jLNW3rJcI07LPPKd0GaVv0U5p5Ue+eQ7MUqqecv0lVLW5ZXvLW27QN9jWF+yC/
hKY1eUrbpg0mAu6aV5y3ai0pWS78RD7racn0rK98o8y1fGeCIs4CtP0SpHr9aRF/BVcUS5CbjoaQh4f/
+yz0fAU2lI+Cr//iUh4Ceyg79YKivYJXRCqUw0vYch2Ja/
hbHUFXAmp0X7yaiZpAjUd6i6iLmI6CiiJGSGrykrBPECdUPmU+OGv5U1fqQUW1H3TceXsix4nMULIaFszRfTn
fpxf4K9/
Xp0lzSjbSIgWG4Yl4e9W9HI7xaf1Q2lLlvm8jxB0bh3nJ9gerpcDEX0pq3z5LhGy81CMhsN02i2xMtNgYiNmQ
CEs4MTplJYnDuVRIDV3oqd6D6j+S36A4Nw53V2Yj88JGMek/0qa/QG1ocB/6Q/
```

```
D0uuCV7JyKuv7SNAXr7G+nwC2T8DJ92rK8mDp9JZUKLGxxFAnB12tq0QdyUZFFsBzwyfSMh/
brtmes6szZlnTF//K3yQFn4wJAwqh4on00bPTXVJBgz65TF3CiLB09lNnEs9QXKST5mMzYoHuiR11f0B/
iSU0YPPCSIyGufRvge7bDaUgFuH845/8QAtxq5ve8KrrjU4A+FDdLUpu5UR4oUBjD8q6twIqKe76AVb1S/
UH2JLrbiP7jcy1Eplyyyifs2Ca6XvHLB1d0wLVdGFhBLRHmbA9ZjCUthBuhGr7QmQ04bJAA8Q7dsQTEJHiIn6
wBJ4WyCrNDkWuQViS605jjgPURUd0dqsEmIPPzGSHiH2k7NDiCyP/
L4MyyNHNteBfYvSDl7ffZtWkTC+eDnPPD4Gds4KI4ENC/nS0m8BvkDPeZs+g88XDD//
YwZ7GYnj46uwuhfNkrzjwSdB33Gbzkb1r1MBMgSdjLLjNqqBv99ApU7Zgy01br+yD+6QnmSe6wuwG+VW1tcCo
6RtpVJzuIRu4+TzxJAymi+YJEABrd05k/
UNBmimai1KMZhRcDE6YG3bZZMABVSLZHTLZtguu98cBt6lLQMqHEggrxX6PCiRhZ+HBPvTgMtv7E4Be5N7
huTH03Drb6xMatZaDusig54rESG8BHnkOKSGw+EK1swz8EJ/
tnwuGaPhen0AAvYldigYKA035Xp+AD0WLNvbPhUwJ3Q4IUDN37J8LbQ3RM0kFHqTzHfQWJKGS5/
QB1AXYSH8sJKHyZAAobmovIZh0Uu7SgXY9hK2/kuoiIyEoCHxX0Jm/u9S036/hL9fW4Aa//
dLWCCbxifQA0LYILvUwZGDeoAWX090AvQYtHuyfC+ULCuQf0rWkLNZSDSbe+VC4wcFV3Q0PLVCCxuFIA00lqrE
28E0c4qVg1NRMaiQd3kQ1l86Vqhb0Alrm0H55agAk/
VcCL0Fzr91NDmWclOnWJ9AKfPBbIGt4qTf5qQhE4WcgVKICrmgKFa4Z5THh9sJdEo+fpLCwq+05aYQgbv3CHn
YwaY5HZ0AcCGp4nSsdgPyo+6tj+FkPEt0SlzjRp0Gowcj/
QSuEGUsjNP0Ds4l0eCcxE7KrWJM7DrwnkfDcHHq+TywRz+g3M0xQMryTUKv+AaYYPBW4oybv0e8Xg4cCdjj1
x4zQE+z02HFYH3EAmMdzgKx48boRl+T3c8CB8nMINug0WNxx0g51CeRTgVoQRz9AopP9wwbLIDeB2Eq4BIGAU
6HWnCM76myFLc3k3SIfcKLceUodiq09k8DpiI9R8pG/
N3WL7wENG0yx5ZppcLLtHZqe5FAefgbWny7hrNwz8B6NBFW0DPyTgBobYwJg5Y5LWBz20Ew32TxGfscLqTuGj
PxrMnyLYJvAX1kAY0oZ4Ysqm7AhAPfsJ3iCT6fidPhDfDrNitnQaBHmlnbYtTsiCgmNe4vzGXvCuLYpT1DDy
whjt5bslRJVyry2Ncf5h00o5gn/
PU8TcqcK270+0PKAstBWJgHm1E8lloyP7JRsjOBPNX6g76KkwbFyk0NgmzIn6xzizVI3VlDrFl81t+k5CXP
7+vptYnANBQp+kLeV0069W2+4SwyOnLNTpg72F/
KL6mN0T63Zq8D8G6cTkYj0xSLtvNJB91RkQw8WVQcfoUh7B3k7cxTgXMkNPdS89mt45JJhdpwz5PKd8iAP/
qpyeH4QH8tA9pL1ZjOr20m2nUESxnGrGrxTrcRhW0t3r05kFb7jAixsAZmPH7u7/
hv1bmDsfCPFvjhnGdMHzwvRlc52PatdRXadtXhJJFj5ocBEvqzVVGpW0t0ojSCa/
zIcFAS8u7M+Bt1RqmmVvnoUQKkIy+0/3piIWo/HUOSCY5y1WrmnZ2/
ZOKF4/3Gci1Lsat5FnuVg2Sw1b3g0uDaIj4WqgJSp98yw1HeZkFn5dvaI3kV2HdKFN3z01QXEZ9GzMTa+DvgE
PUDMgPBM7etoGm+qs+E+aandKfuSj4Ck5D4Lq1z26IkALbPhoanUm3MFHftUPLoGgUxsZ6Gs5PbeQH6fznUHW
13giV9CLpYrPAJ0WmJgnOdH+Bf4agFnVLv+byBqgnPerWZNRdBV5LkLyJlELPS21g1c3EQu9KYXIMJhD5F4Y
viTYD7WGOZKQ/
bApGILvbytdb0sSjSQNCbZ0ADu7vMnJ3qDCMCFWYvg7f0nwWtojcCKp3+hrAaqKpD+6kEWC+dATrXLBkTozey
ULHNo5mAXZVIRECqMctSFsWfQxKGDDihksp8UcyP8C8Bg+KvF8fevgY4wZHEuNgCEu7wxsgjjWYGFJ8gSm4pC
jqLpkP+W300nz1Vh+LFAnn1EPb5mlzMcBMLG2bx+LaMufUrD7FXkFihyTgsTI3KKTHdBMLTyrnLJoxxuZb0+
83Ngw+hfyStxt8EsSjzSrsVjoGagM44/6Rj9h5eSc1K9IuRbbaD6w0ULbxtAtFCLJxNKQoq+qY6v5yYg6Az
FseASgJOW35CbZ6dEhAwM+UH+xC0SsAp1+iMV4RnQdCL5n1cVbgduVoUptnDkLub/
wZr+nBveok7LqW5XB9rL0CQiR/VKTVHhShHciC98/vpP1/TnhGcmavsnurww38fjuw/
8LyBfpz0WqmSNj880f+c32GrXiN7DGtMfKekZJHsTubllWZeTiI1iVVVyK62Wv1vAu7uhtsf7B9Iso95egFgc
AAAAAE1FTkSuQmCC");
```

### 7.6.7 leaflet.marker.icon.[].iconSize

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Größe des Iconbildes in Pixeln.

```
dashlet.addOption("leaflet.marker.icon.1.iconSize","[24, 24]");
```

### 7.6.8 leaflet.marker.icon.[]iconAnchor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: null

Die Koordinaten der "Spitze" des Symbols (relativ zu seiner oberen linken Ecke). Das Symbol wird so ausgerichtet, dass sich dieser Punkt an der geografischen Position des Markers befindet. Standardmäßig zentriert, wenn Größe angegeben ist, kann auch in CSS mit negativen Rändern festgelegt werden.

```
dashlet.addOption("leaflet.marker.icon.1.iconAnchor","[12, 12]");
```

### 7.6.9 leaflet.marker.icon.[]popupAnchor

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: [0, 0]

Die Koordinaten des Punkts, von dem aus Popups "geöffnet" werden, relativ zum Symbolanker.

```
dashlet.addOption("leaflet.marker.icon.1.popupAnchor", "[-3, -76]");
```

### 7.6.10 leaflet.marker.icon.[]shadowUrl

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: null

Die URL zum Symbolschattenbild. Wenn nicht angegeben, wird kein Schattenbild erstellt.

```
dashlet.addOption("leaflet.marker.icon.1.shadowUrl", "");
```

### 7.6.11 leaflet.marker.icon.[]shadowSize

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: null

Größe des Schattenbildes in Pixel.

```
dashlet.addOption("leaflet.marker.icon.1.shadowSize", "");
```

### 7.6.12 leaflet.marker.icon.[]shadowAnchor

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: null

Die Koordinaten der "Spitze" des Schattens (relativ zu seiner oberen linken Ecke) (wie iconAnchor, falls nicht angegeben).

```
dashlet.addOption("leaflet.marker.icon.1.shadowAnchor","");
```

### 7.6.13 leaflet.marker.icon.active2

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Setzt den Marker auf das entsprechende Icon, wenn er ausgewählt ist.

Hinweis: Die Details des Icons müssen mithilfe der Icon-Optionen separat angegeben werden.

```
dashlet.addOption("leaflet.marker.icon.active2","leaflet1_icon2");
```

### 7.6.14 leaflet.marker.event.click

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>

<b>Verfügbar seit Version</b>	2.2.8
<b>Schlagworte / Tags</b>	

Funktion für den Click-Event

```
dashlet.addOption("leaflet.marker.event.click", "alert('Click-Test');");
```

### 7.6.15 leaflet.markers.autoload

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Marker
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.3.8
<b>Schlagworte / Tags</b>	

Default: false

Option zum Anzeigen aller Markers beim Rendern des Dashlets

```
dashlet.addOption("leaflet.markers.autoload", "true");
```

## 8 OrgChart [Options]

<b>Name</b>	OrgChart
<b>Version</b>	2.1.9
<b>Hersteller</b>	-
<b>Webseite</b>	<a href="https://github.com/dabeng/OrgChart">https://github.com/dabeng/OrgChart</a>

### 8.1 OrgChart General

#### 8.1.1 orgchart.all.options

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit die Optionen als JSON formatierten String direkt/manuell zu übergeben.

## 9 PivotTable [Options]

<b>Name</b>	PivotTable
<b>Version</b>	2
<b>Hersteller</b>	Nicolas Kruchten
<b>Webseite</b>	<a href="https://pivottable.js.org/examples/">https://pivottable.js.org/examples/</a>

### 9.1 PivotTable General

#### 9.1.1 pivot.custom.script.before

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit um bei der PivotTable ein Script vor der Initialisierung einzufügen.

#### 9.1.2 pivot.display.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Setzt den Display Typ der PivotTable Komponente.

Akzeptierte Werte: "pivotUI" oder "pivot"



### 9.1.3 pivot.all.options

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit die Optionen als JSON formatierten String direkt/manuell zu übergeben.

### 9.1.4 pivot.rows

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

Array von Attributnamen, die als Zeilen verwendet werden.

### 9.1.5 pivot.cols

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

Array von Attributnamen, die als Spalten verwendet werden.

### 9.1.6 pivot.agggregator

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.pivotUtilities.aggregators["Count"]() ]

Konstruktor für ein Objekt, welches Ergebnisse pro Zelle aggregiert.

### 9.1.7 pivot.agggregatorName

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "Count" ]

pivot(): Name des Aggregators, in manchen Renderern für Anzeigezwecke verwendet.

pivotUI(): Aggregator, mit dem das Dropdown vorbefüllt werden soll i.e. Schlüssel zum aggregators-Objekt.

### 9.1.8 pivot.renderer

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: table() ]

Generiert output aus einer Pivot-Datenstruktur.

### 9.1.9 pivot.rowOrder

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "key\_a\_to\_z" ]

Die Reihenfolge in der Zeilendaten an den Renderer übergeben werden. Muss "key\_a\_to\_z", "value\_a\_to\_z" oder "value\_z\_to\_a" sein. Nach Wert zu sortieren sortiert nach Zeilensumme.

### 9.1.10 pivot.colOrder

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: "key\_a\_to\_z" ]

Die Reihenfolge in der Spaltendaten an den Renderer übergeben werden. Muss "key\_a\_to\_z", "value\_a\_to\_z" oder "value\_z\_to\_a" sein. Nach Wert zu sortieren sortiert nach Spaltensumme.

### 9.1.11 pivot.derivedAttributes

<b>Datentyp</b>	FUNCTION
-----------------	----------

<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: {} ]

Objekt um abgeleitete Attribute zu definieren.

### 9.1.12 pivot.dataClass

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.pivotUtilities.PivotData ]

Konstruktor für die Datenklasse die gebaut und an den Renderer übergeben werden soll (sollte eine Subklasse der Standardoption sein).

### 9.1.13 pivot.filter

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: function(){return true;} ]

Für jeden Eintrag aufgerufen, gibt false zurück falls der Eintrag vor dem Rendering aus dem Input ausgeschlossen werden soll, ansonsten true.

### 9.1.14 pivot.sorters

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: {} ]

Wird mit einem Attributnamen aufgerufen, beziehungsweise darauf zugegriffen, und kann eine Funktion zurückgeben, die zu Outputzwecken als Argument an `array.sort` übergeben werden kann. Wird keine Funktion zurückgegeben, ist der Standard-Sortiermechanismus eine eingebaute "natural sort"-Implementierung.

### 9.1.15 pivot.rendererOptions

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: {} ]

Objekt, welches als Optionen an den Renderer durchgereicht wird.

### 9.1.16 pivot.localeStrings

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5

<b>Schlagworte / Tags</b>	
---------------------------	--

[ Default-Wert: en strings ]

Locale-spezifische Strings für Fehlermeldungen.

### 9.1.17 pivot.vals

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

Attributnamen die in den vals Bereich voreingefügt werden sollen (wird an Aggregator generierende Funktion weitergereicht).

### 9.1.18 pivot.aggregators

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.pivotUtilities.aggregators ]

Lexikon an Generatoren für Aggregationsfunktionen im Dropdown.

### 9.1.19 pivot.renderers

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: \$.pivotUtilities.renderers ]

Lexikon an Renderfunktionen.

### 9.1.20 pivot.inclusions

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: {} ]

Objekt, dessen Schlüssel Attributnamen sind und dessen Werte Arrays von Attributswerten sind, die bezeichnen, welche Einträge beim Rendern miteinbezogen werden sollen. Verwendet um die Filtermenüs vorzubefüllen die bei einem Doppelklick erscheinen (überschreibt exclusions).

### 9.1.21 pivot.exclusions

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: {} ]

Objekt, dessen Schlüssel Attributnamen sind und dessen Werte Arrays von Attributswerten sind, die bezeichnen, welche Einträge beim Rendern ausgeschlossen werden sollen. Verwendet um die Filtermenüs vorzubefüllen die bei einem Doppelklick erscheinen.

### 9.1.22 pivot.hiddenAttributes

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

Enthält Attributnamen, die in der UI weggelassen werden sollen.

### 9.1.23 pivot.hiddenFromAggregators

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: [] ]

Enthält Attributnamen, die in den Aggregator-Dropdowns weggelassen werden sollen.

### 9.1.24 pivot.hiddenFromDragDrop

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	



[ Default-Wert: [] ]

Enthält Attributnamen, die im drag'n'drop-Teil der UI wegelassen werden sollen.

### 9.1.25 pivot.onRefresh

<b>Datentyp</b>	FUNCTION
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	2.2.5

[ Default value: function({}) ]

Bei Aktualisierung des Renderers aufgerufen mit einem Objekt, welches die momentanen UI-Einstellungen repräsentiert.

### 9.1.26 pivot.menuLimit

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 50 ]

Maximale Anzahl an Werten die im Doppelklick-Menü angezeigt werden.

### 9.1.27 pivot.autoSortUnusedAttrs

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: false ]

Kontrolliert, ob unbenutzte Attribute in der UI sortiert gehalten werden, oder nicht.

### 9.1.28 pivot.unusedAttrsVertical

<b>Datentyp</b>	UNKNOWN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: 85 ]

Kontrolliert, ob unbenutzte Attribute vertikal angezeigt werden statt wie standardmäßig horizontal. Der Wert true bedeutet immer vertikal, der Wert false bedeutet immer horizontal. Falls der Wert auf eine Zahl gesetzt ist (wie in der Standardeinstellung), werden die Attribute nur vertikal angezeigt, falls die Länge ihrer kombinierten Namen in Zeichen größer ist als der angegebene Wert.

### 9.1.29 pivot.showUI

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default-Wert: true ]

Kontrolliert, ob die drag'n'drop UI angezeigt werden soll. Auf false stellen, um das Verhalten von pivot() mit der Signatur von pivotUI() zu emulieren.

### 9.1.30 pivot.rendererName

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

[ Default value: Erster Schlüssel in renderers ]

Renderer, mit dem das Dropdown vorbefüllt werden soll i.e. Schlüssel zum renderers-Objekt.

### 9.1.31 pivot.showSubTotal

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Default Wert: false

Möglichkeit zum Rendern der Pivot-Daten mit Untersumme.

Der Optionsblock sollte wie im folgenden die Option 'dataClass' enthalten.

```
dashlet.addOption("pivot.all.options", "renderers: renderers,\n" +
    " dataClass: dataClass,\n" + /* required for subtotal!!*/
    "cols: [\"Postal Code\", \"Creditlimit\"],\n" +
    "rows: [\"City\", \"Customer Name\"],\n" +
    "rendererName: \"Table With Subtotal Heatmap\",\n" +
    "aggregatorName: \"Maximum\", \n" + // \"Average\"
    "vals: [\"Creditlimit\"], \n" +
    "rowOrder: \"value_z_to_a\", colOrder: \"value_z_to_a\","+
    "  rendererOptions: {\n"+
    "    c3: {\n"+
    "      size: { width:1040, height:650 } \n"+
    "    } \n"+
    "  }"
);
```

### 9.1.32 pivot.add.buttonbar

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Fügt eine Buttonbar für den Seriesfilter hinzu

Example:

```
// Die Spalte Country wird als Series-Filer definiert
dashlet.addOption("datatable.column.seriesfilter", "Country");
```

```
dashlet.addOption("pivot.add.buttonbar", "seriesfilter");
```

### 9.1.33 pivot.load.json.script.success

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, das auszuführende Skript zu übergeben, wenn der externe JSON-Aufruf erfolgreich ist.

Dies wird verwendet, wenn die Daten für die Pivot-Tabelle in einer externen JSON-Datei gespeichert werden.

### 9.1.34 pivot.load.json.script.error

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, das auszuführende Skript zu übergeben, wenn der externe JSON-Aufruf fehlgeschlagen ist.

Dies wird verwendet, wenn die Daten für die Pivot-Tabelle in einer externen JSON-Datei gespeichert werden.

### 9.1.35 pivot.load.json.script.done

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, das auszuführende Skript zu übergeben, wenn der externe JSON-Aufruf fertig ist.

Dies wird verwendet, wenn die Daten für die Pivot-Tabelle in einer externen JSON-Datei gespeichert werden.

### 9.1.36 pivot.load.json.script.complete

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, das auszuführende Skript zu übergeben, wenn der externe JSON-Aufruf abgeschlossen ist.

Dies wird verwendet, wenn die Daten für die Pivot-Tabelle in einer externen JSON-Datei gespeichert werden.

### 9.1.37 pivot.json.path

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit, den Pfad zur externen JSON-Datei zu übergeben.

Dies wird verwendet, wenn die Daten für die Pivot-Tabelle in einer externen JSON-Datei gespeichert werden.

Example:

```
// set the path for the external JSON files (only for servBIRD)
glJobID = 0;
var appContext = reportContext.getAppContext(); // HashMap
if (appContext != null && appContext.containsKey("REPORT_JOBID"))
{
    glJobID = appContext.get("REPORT_JOBID");
}
dashlet.addOption("pivot.json.path", "http://localhost:8080/servBIRD/
streamOutputFile?jobid=" + glJobID + "&filename=d/");
```

### 9.1.38 pivot.custom.script.after

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	General
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>TESTED</b>
<b>Verfügbar seit Version</b>	2.2.5
<b>Schlagworte / Tags</b>	

Möglichkeit um bei der PivotTable ein Script nach der Initialisierung einzufügen.

## 10 Spreadsheet [Options]

<b>Name</b>	Spreadsheet
<b>Version</b>	8.1.1
<b>Hersteller</b>	Paramvir Dhindsa
<b>Webseite</b>	<a href="http://paramquery.com">http://paramquery.com</a>

### 10.1 Spreadsheet Column

#### 10.1.1 spreadsheet.column.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Column-Komponente

Beispiel:

```
dashlet.addOption("spreadsheet.column.object" , "["
    + "{title: 'Productline' ,dataIndx : 'Productline' },"
    + "{title: 'Code' ,dataIndx : 'Code' },"
    + "{title: 'Name' ,dataIndx : 'Name' },"
    + "{title: 'Scale' ,dataIndx : 'Scale' },"
    + "{title: 'Vendor' ,dataIndx : 'Vendor' },"
    + "{title: 'MSRP price' ,dataIndx : 'MSRP price' },"
    + "{title: 'Buy price' ,dataIndx : 'Buy price' },"
    + "{title: 'Quantity in stock' ,dataIndx : 'Quantity in stock' },"
    + "{title: 'Sales' ,dataIndx : 'Sales' },,];"
    + "]" );
```

#### 10.1.2 spreadsheet.column.n[.align

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der Ausrichtung der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].align", "right");
```

### 10.1.3 spreadsheet.column.n[].dataType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der DataTyp der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].dataType", "float");
dashlet.addOption("spreadsheet.column.n[Code].dataType", "string");
```

### 10.1.4 spreadsheet.customColumn.i[].format

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16



### Schlagworte / Tags

Option zum Einstellen der Format der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[0].format", "##,###.00 €");
dashlet.addOption("spreadsheet.customColumn.i[1].format", "yy-mm-dd");
dashlet.addOption("spreadsheet.customColumn.i[2].format", "percent"); // or %
```

## 10.1.5 spreadsheet.column.n[].hidden

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der Sichtbarkeit der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].hidden", "true");
```

## 10.1.6 spreadsheet.column.n[].render.object

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Renderfunktion für eine Spalte. Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.object", "function (ui) {n" +
"  if (ui.rowData.Sales > 100000) { return {text: "<img src='arrow_up.png'/>&nbsp;";
+ new Intl.NumberFormat('de-DE', { style: 'currency', currency:
'EUR' }).format(ui.rowData.Sales),n" +
"    style: { 'background': '#00ff00'} }; n" +
"  }else { return {text: "<img src='arrow_down.png'/>&nbsp;"; + new
Intl.NumberFormat('de-DE', { style: 'currency', currency:
'EUR' }).format(ui.rowData.Sales),n" +
"    style: { 'background': '#0000ff'} }; }" +
"} ");
```

### 10.1.7 spreadsheet.column.n[.render.condition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Renderfunktion-Kondition für eine Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.condition", "Sales > 100000");
```

### 10.1.8 spreadsheet.column.n[.render.conditionTrue.style

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der RenderStyle für eine Spalte falls der Konidtion True wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionTrue.style", "'font-style': 'italic' , 'color': '#00ff00' ");
```

### 10.1.9 spreadsheet.column.n[].render.conditionFalse.style

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Renderfunktion-style für eine Spalte, falls es False wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionFalse.style", "'font-style': 'italic' , 'color': '#00ff00' ");
```

### 10.1.10 spreadsheet.column.n[].render.conditionTrue.style.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Render-FontStyle für eine Spalte falls der Konidtion True wird

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionTrue.style.fontStyle", "italic");
```

### 10.1.11 spreadsheet.column.n[].render.conditionFalse.style.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Render-FontStyle für eine Spalte falls der Konidtion False wird

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionFalse.style.fontStyle", "italic");
```

### 10.1.12 spreadsheet.column.n[].render.conditionTrue.style.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Render-Color für eine Spalte falls der Konidtion True wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionTrue.style.color", "#00ff00");
```

### 10.1.13 spreadsheet.column.n[].render.conditionFalse.style.color

<b>Datentyp</b>	COLOR
-----------------	-------

<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Render-Color für eine Spalte falls der Konidtion false wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].render.conditionFalse.style.color", "#00ff00");
```

#### 10.1.14 spreadsheet.column.n[].aggregate.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Summaryfunktion für eine Spalte.

predefined aggregates sind "avg", "count", "min", "max", "sum"

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].aggregate.type", "sum");
```

#### 10.1.15 spreadsheet.column.n[].aggregate.edit

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Default: true

Option zum Festlegen der Edit-funktion für eine Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].aggregate.edit", "true");
```

### 10.1.16 spreadsheet.column.n[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der Title der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].title", "Sale");
```

### 10.1.17 spreadsheet.customColumn.i[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Titels der benutzerdefinierten Spalten. Die Indizes der benutzerdefinierten Spalten sollten berücksichtigt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[0].title", "2015 (Q4-Q3)");
dashlet.addOption("spreadsheet.customColumn.i[1].title", "2015 (Q3-Q2)");
```

### 10.1.18 spreadsheet.customColumn.i[].value

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Werts der benutzerdefinierten Spalten. Die Indizes der benutzerdefinierten Spalten sollten berücksichtigt werden. Der Wert ist einzugeben als `<erster_operand>;<zweiter_operant>;<operation>`

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[0].value", "NETTOWERT;MENGE;/");
```

### 10.1.19 spreadsheet.customColumn.i[].aggregate.type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Summaryfunktion für eine Custom-Spalte.

predefined aggregates sind "avg", "count", "min", "max", "sum"

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[0].aggregate.type", "sum");
```

### 10.1.20 spreadsheet.customColumn.i[].dataType

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der DataTyp der custom-Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[0].dataType", "integer");
```

### 10.1.21 spreadsheet.customColumn.i[].value.object

<b>Datentyp</b>	SCRIPT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Value-Object der benutzerdefinierten Spalten. Die Indizes der benutzerdefinierten Spalten sollten berücksichtigt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.customColumn.i[2].value.object", "return ((new Date (rd.BELEGDATUM).getMonth()+1) + '-' + new Date (rd.BELEGDATUM).getFullYear()));");
dashlet.addOption("spreadsheet.customColumn.i[2].title", "BELEGDATUM_group");
```



### 10.1.22 spreadsheet.column.n[].render.condition.sameColumn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Renderfunktionsbedingung, wenn die Spalte in der Bedingung gleich ist.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].render.condition.sameColumn",
">1000");
```

### 10.1.23 spreadsheet.column.n[].format

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Einstellen der Format der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].format", "##,###.00 €");
dashlet.addOption("spreadsheet.column.n[DATUM].format", "yy-mm-dd");
dashlet.addOption("spreadsheet.column.n[Test].format", "percent"); // or %
```

### 10.1.24 spreadsheet.column.n[].clsHead

<b>Datentyp</b>	STRING
-----------------	--------

<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Klasse, die nur der Kopfzeile zugewiesen wird.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].clsHead", "bold_class");
```

### 10.1.25 spreadsheet.column.n[].cls

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Klasse, die der gesamten Spalte einschließlich der Überschrift zugewiesen werden soll.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].cls", "bold_class");
```

### 10.1.26 spreadsheet.column.n[].tpCls

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Klasse, die einer im ToolPanel angezeigten Spalte zugewiesen werden soll. Es ist nützlich, einer Spalte CSS-Stile wie Hintergrundfarbe, Schriftart usw. zuzuweisen. Es gibt eine spezielle Klasse "pq-deny-drag", die, wenn sie einer Spalte zugewiesen wird, verhindert, dass sie in toolPanel gezogen wird.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].tpCls", "red");
```

### 10.1.27 spreadsheet.column.n[].halign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Er bestimmt die horizontale Ausrichtung des Textes in den Spaltenüberschriften.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].halign", "right");
```

Erlaubte Werte

- left
- right
- center

### 10.1.28 spreadsheet.column.n[].hvalign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Definiert die vertikale Ausrichtung des Textes in den Spaltenüberschriftenzellen

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].halign", "top");
```

### Erlaubte Werte

- center
- top
- bottom

## 10.1.29 spreadsheet.column.n[].valign

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Es definiert die vertikale Ausrichtung des Textes in den Spalten.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].valign", "top");
```

### Erlaubte Werte

- center
- top
- bottom

## 10.1.30 spreadsheet.column.n[].width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Breite der Spalte in Pixeln oder in Prozent der Breite des Rasters. Standardmäßig wird die Mindestbreite der Spalte verwendet, wenn die Breite nicht angegeben ist oder wenn die Breite kleiner als die Mindestbreite ist.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].width", "100");
```

```
dashlet.addOption("spreadsheet.column.n[Country].width", "50%");
```

### 10.1.31 spreadsheet.column.n[].maxWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Maximal mögliche Breite der Spalte in Pixeln oder in Prozent der Breite des Rasters.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].maxWidth", "100");
```

```
dashlet.addOption("spreadsheet.column.n[Country].maxWidth", "50%");
```

### 10.1.32 spreadsheet.column.n[].minWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Minimal mögliche Breite der Spalte in Pixeln oder in Prozent der Breite des Rasters.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].minWidth", "100");
```

```
dashlet.addOption("spreadsheet.column.n[Country].minWidth", "50%");
```

### 10.1.33 spreadsheet.column.n[].menuIcon

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wird verwendet, um die Anzeige des Kopfzeilenmenüsymbols für diese Spalte anzuzeigen oder zu überschreiben.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].menuIcon", "true");
```

### 10.1.34 spreadsheet.column.n[].menuUI

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wird verwendet, um die Kopfzeilenmenüstruktur für diese Spalte zu definieren oder zu überschreiben

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].menuUI", "true");
```

### 10.1.35 spreadsheet.column.n[].menuInDisable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wenn dies der Fall ist, ist die Spalte im Kopfzeilenmenü deaktiviert, sodass der Status der Checkbox nicht geändert werden kann.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].menuInDisable", "true");
```

### 10.1.36 spreadsheet.column.n[].menuInClose

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wenn dies der Fall ist, wird die gruppierte Spalte im Kopfzeilenmenü reduziert.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].menuInClose", "true");
```

### 10.1.37 spreadsheet.column.n[].menuInHide

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Spalte wird im Kopfzeilenmenü ausgeblendet, wenn dies gesetzt wird.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].menuInHide", "true");
```

### 10.1.38 spreadsheet.column.n[].nodrag

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	IMPLEMENTED
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Spalte kann nicht gezogen werden, wenn dies gesetzt ist.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].nodrag", "true");
```

### 10.1.39 spreadsheet.column.n[].nodrop

<b>Datentyp</b>	BOOLEAN
-----------------	---------



<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wenn diese Option auf "true" gesetzt ist, kann keine ziehbare Spalte auf dieser Spalte abgelegt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].nodrop", "true");
```

#### 10.1.40 spreadsheet.column.n[].resizable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Die Größe der Spalte kann je nach Wert dieser Option geändert werden.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].resizable", "true");
```

#### 10.1.41 spreadsheet.column.n[].sortable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Setz der Option auf false, um die Sortierung für eine Spalte zu deaktivieren.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].sortable", "true");
```

### 10.1.42 spreadsheet.column.n[].type

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Option zum Einstellen der typ der Spalte als 'checkbox' or 'detail'

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].type", "detail");
dashlet.addOption("spreadsheet.column.n[Code].type", "checkbox");
```

Erlaubte Werte

- checkbox
- detail

### 10.1.43 spreadsheet.column.n[].tpHide

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19

### Schlagworte / Tags

Blenden Sie diese Spalte im ToolPanel aus.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].tpHide", "true");
```

## 10.1.44 spreadsheet.column.n[].style

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Definiert den CSS-Stil für alle Zellen in der Spalte

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].style", "'font-style': 'italic' , 'color': '#00ff00' ");
```

## 10.1.45 spreadsheet.column.n[].styleHead

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Definiert den CSS-Stil nur für die Spaltenüberschriftszelle.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[Sales].styleHead", "'font-style': 'italic' ,  
'color': '#00ff00' ");
```

### 10.1.46 spreadsheet.column.n[].aggregate.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Column
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Ein Objekt, das Typ- und Edit-eigenschaften des Aggregats für diese Spalte enthält.

Beispiel:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].aggregate.object", "{ type: \"min\",  
edit: true}");
```

## 10.2 Spreadsheet Group Model

### 10.2.1 spreadsheet.groupModel.object

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der GroupModel-Komponente

Beispiel:

```
dashlet.addOption ("spreadsheet.groupModel.object" , "{n" +
"      on: true,n" +
"      dataIndx: ['Productline'],n" +
"      collapsed: [false],n" +
"      title: [n" +
"        \"{0} ({1})\",n" +
"        \"{0} - {1}\"n" +
"      ]n" +
"    }");
```

### 10.2.2 spreadsheet.groupModel.dataIndx

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Spalten, die gruppiert werden sollen

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.dataIndx", "'Productline', 'Scale'");
```

### 10.2.3 spreadsheet.groupModel.collapsed

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Default: False

Option zum Definieren, ob die gruppierten Spalten aufgeklappt werden sollen oder nicht.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.collapsed", "false");
```

## 10.2.4 spreadsheet.groupModel.grandSummary

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Default: False

Option zum Definieren, ob die GrandSummary der gruppierten Spalten darstellt werden sollen oder nicht.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.grandSummary", "false");
```

## 10.2.5 spreadsheet.groupModel.agg

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

es ist ein Objekt von dataIdx: Paare von aggregierten Spalten ein, z. B. { silver: 'min', gold: 'sum' }

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.agg", "{ silver: 'min', gold: 'sum' }");
```

## 10.2.6 spreadsheet.groupModel.fixCols

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Die groupby-Spalten werden in der gleichen Reihenfolge wie groupModel.dataIdx nach links vom Ansichtsfenster verschoben und fixiert, wenn diese Option auf true gesetzt ist.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.fixCols", "false");
```

## 10.2.7 spreadsheet.groupModel.groupCols

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

ein Array von dataIdx zum Gruppieren entlang der Spaltenachse, d. h. ['Jahr', 'Sport']

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.groupCols", "{ silver: 'min', gold: 'sum' }");
```

## 10.2.8 spreadsheet.groupModel.header

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Die Kopfzeilen-Symbolleiste wird angezeigt, um Spalten hinzuzufügen (mit Drag/Drop) oder aus der Gruppierung zu entfernen und auch um die Reihenfolge der gruppierten Spalten neu anzuordnen.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.header", "true");
```

### 10.2.9 spreadsheet.groupModel.headerMenu

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Das Menü wird in der Symbolleiste der Gruppierungskopfzeile angezeigt, um boolesche Gruppenmodell-Unteroptionen durch den Endbenutzer umzuschalten.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.headerMenu", "true");
```

### 10.2.10 spreadsheet.groupModel.icon

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Symbole, die vor dem Titel der Gruppe angezeigt werden. Der erste Wert im Array entspricht dem Symbol des erweiterten Zustands und der zweite dem reduzierten Zustand.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.icon", "[ 'ui-icon-triangle-1-se', 'ui-icon-triangle-1-e' ]");
```

### 10.2.11 spreadsheet.groupModel.indent

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Öffnen/Schließen-Symbole in Gruppentiteln werden von links um (Einzug \* Ebene) Pixel eingerückt, wobei Ebene die Gruppierungsebene der groupby-Spalte ist.

### 10.2.12 spreadsheet.groupModel.menuItems

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Boolesche Optionen, die im Gruppierungskopfzeilenmenü angezeigt werden. Weitere boolesche Optionen können hinzugefügt oder entfernt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.menuItems", "[ "merge", "fixCols", "grandSummary" ]");
```

### 10.2.13 spreadsheet.groupModel.merge

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Die gruppierten Zellen mit demselben Wert werden vertikal zusammengeführt, wenn diese Option wahr ist.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.merge", "false");
```

### 10.2.14 spreadsheet.groupModel.nodeClose

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Abrufen oder Festlegen des `pq_close`-Werts aller übergeordneten Knoten im Zeilengruppierungs- oder Pivot-Modus. Diese Option ist nützlich bei der Anfangs- oder Laufzeitverwaltung des geöffneten/geschlossenen Zustands von Knoten.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.nodeClose", "{ Argentina: false, // Argentina is the id of the node and false means node is open Argentina_Yvonne Moncada: true, Argentina_Sergio Gutiérrez: true, ... }");
```

### 10.2.15 spreadsheet.groupModel.on

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Option zum Ein- und Ausschalten der Gruppierung.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.collapsed", "false");
```

### 10.2.16 spreadsheet.groupModel.pivot

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Group Model
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Wenn true, wird der Pivot-Modus aktiviert.

Beispiel:

```
dashlet.addOption("spreadsheet.groupModel.pivot", "false");
```

## 10.3 Spreadsheet Spreadsheet Object

### 10.3.1 spreadsheet.rowInit.object

<b>Datentyp</b>	SCRIPT
-----------------	--------

<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit funktion. Diese Funktion definiert das Zeilen-Styling basierend auf einer Bedingung. Hinweis: Nur Hex-Code kann in der Export benutzt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.object", "function (ui) {\n" +
"         if (ui.rowData.Sales < 100000) { return { style: {'font-\n" +
"style': 'italic' , 'color': '#00ff00' } }; }\n" +
"         }");
```

### 10.3.2 spreadsheet.rowInit.condition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit Funktion-Kondition

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.condition", "Sales > 100000");
```

### 10.3.3 spreadsheet.rowInit.conditionTrue.style

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content

<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit Style falls der Konidtion True wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionTrue.style", "'font-style': 'italic' , 'color': '#00ff00' ");
```

### 10.3.4 spreadsheet.rowInit.conditionFalse.style

<b>Datentyp</b>	OBJECT
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit Funktion-style , falls es False wird. Hinweis: Nur Hex-Code kann in der Export benutzt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionFalse.style", "'font-style': 'italic' , 'color': '#00ff00' ");
```

### 10.3.5 spreadsheet.rowInit.conditionTrue.style.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>

<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit-FontStyle falls der Konidtion True wird

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionTrue.style.fontStyle", "italic");
```

### 10.3.6 spreadsheet.rowInit.conditionFalse.style.fontStyle

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit-FontStyle falls der Konidtion False wird

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionFalse.style.fontStyle", "italic");
```

### 10.3.7 spreadsheet.rowInit.conditionTrue.style.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit-Color falls der Konidtion True wird.Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionTrue.style.color", "#00f00");
```

### 10.3.8 spreadsheet.rowInit.conditionFalse.style.color

<b>Datentyp</b>	COLOR
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der rowInit-Color falls der Konidtion false wird.Hinweis: Nur Hex-Code kann in der Export benutzt werden,

Beispiel:

```
dashlet.addOption("spreadsheet.rowInit.conditionFalse.style.color", "#00ff00");
```

### 10.3.9 spreadsheet.pageModel.enable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der Paging

Beispiel:

```
dashlet.addOption("spreadsheet.pageModel.enable", "true");
```

### 10.3.10 spreadsheet.pivot.customColumn.i[].title

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Titels der benutzerdefinierten Spalten in einem Pivot. Die Indizes der benutzerdefinierten Spalten sollten berücksichtigt werden.

Beispiel:

```
dashlet.addOption("spreadsheet.pivot.customColumn.i[0].title", "2015 (Q4-Q3)");
dashlet.addOption("spreadsheet.pivot.customColumn.i[1].title", "2015 (Q3-
Q2)");
dashlet.addOption("spreadsheet.pivot.customColumn.i[2].title", "2015 (Q2-
Q1)");
```

### 10.3.11 spreadsheet.pivot.customColumn.i[].value

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Werts der benutzerdefinierten Spalten in einem Pivot. Die Indizes der benutzerdefinierten Spalten sollten berücksichtigt werden. Der Wert ist einzugeben als `<erster_operand>;<zweiter_operant>;<operation>`

Beispiel:



```
dashlet.addOption("spreadsheet.pivot.customColumn.i[0].value",
"2015_3_aggr_NETTOWERT;2015_2_aggr_NETTOWERT;-");
    dashlet.addOption("spreadsheet.pivot.customColumn.i[1].value",
"2015_2_aggr_NETTOWERT;2015_1_aggr_NETTOWERT;-");
    dashlet.addOption("spreadsheet.pivot.customColumn.i[2].value",
"2015_1_aggr_NETTOWERT;2015_0_aggr_NETTOWERT;-");
```

### 10.3.12 spreadsheet.pivot.groupCols.dateColumn

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Namens der Spalte, die als DATE-Spalte berücksichtigt werden soll.

Beispiel:

```
dashlet.addOption("spreadsheet.pivot.groupCols.dateColumn", "BELEGDATUM");
```

Wird **in** Kombination verwendet, wenn die Datumsspalte als Jahr/Quartal/Monat angezeigt werden soll.

```
dashlet.addOption("spreadsheet.pivot.groupCols.year.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.quarter.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.month.enable", "true");
```

### 10.3.13 spreadsheet.pivot.groupCols.year.enable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option, um zu entscheiden, ob die Datumsspalte als Jahr angezeigt werden soll oder nicht

Beispiel:

```
dashlet.addOption("spreadsheet.pivot.groupCols.year.enable", "true");
```

Wird **in** Kombination verwendet, wenn die Datumsspalte als Jahr/Quartal/Monat angezeigt werden soll.

```
dashlet.addOption("spreadsheet.pivot.groupCols.dateColumn", "BELEGDATUM");
dashlet.addOption("spreadsheet.pivot.groupCols.year.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.quarter.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.month.enable", "true");
```

### 10.3.14 spreadsheet.pivot.groupCols.quarter.enable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option, um zu entscheiden, ob die Datumsspalte als Quartal angezeigt werden soll oder nicht

Beispiel:

```
dashlet.addOption("spreadsheet.pivot.groupCols.quarter.enable", "true");
```

Wird **in** Kombination verwendet, wenn die Datumsspalte als Jahr/Quartal/Monat angezeigt werden soll.

```
dashlet.addOption("spreadsheet.pivot.groupCols.dateColumn", "BELEGDATUM");
dashlet.addOption("spreadsheet.pivot.groupCols.year.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.quarter.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.month.enable", "true");
```

### 10.3.15 spreadsheet.pivot.groupCols.month.enable

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option, um zu entscheiden, ob die Datumsspalte als Monat angezeigt werden soll oder nicht

Beispiel:

```
dashlet.addOption("spreadsheet.pivot.groupCols.month.enable", "true");
```

Wird **in** Kombination verwendet, wenn die Datumsspalte als Jahr/Quartal/Monat angezeigt werden soll.

```
dashlet.addOption("spreadsheet.pivot.groupCols.dateColumn", "BELEGDATUM");
dashlet.addOption("spreadsheet.pivot.groupCols.year.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.quarter.enable", "true");
dashlet.addOption("spreadsheet.pivot.groupCols.month.enable", "true");
```

### 10.3.16 spreadsheet.aggregate.i[].name

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Namens der Aggregatfunktion.

Beispiel :

```
dashlet.addOption("spreadsheet.aggregate.i[0].name" , "Sum_Count");
dashlet.addOption("spreadsheet.aggregate.i[0].definition" , "(agg.sum(arr, col) / agg.count(arr, col))");
```

Dieses Aggregat kann dann wie folgt für eine Spalte verwendet werden:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].summary.type", "all");
```

### 10.3.17 spreadsheet.aggregate.i[[]].definition

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen des Namens der Aggregatfunktion.

Beispiel :

```
dashlet.addOption("spreadsheet.aggregate.i[0].name" , "Sum_Count");
dashlet.addOption("spreadsheet.aggregate.i[0].definition" , "(agg.sum(arr, col) / agg.count(arr, col))");
```

Dieses Aggregat kann dann wie folgt für eine Spalte verwendet werden:

```
dashlet.addOption("spreadsheet.column.n[NETTOWERT].summary.type", "all");
```

### 10.3.18 spreadsheet.toolbar.enable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der toolbar

Beispiel:

```
dashlet.addOption("spreadsheet.toolbar.enable", "true");
```

### 10.3.19 spreadsheet.toolPanel.show

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der toolPanel

Beispiel:

```
dashlet.addOption("spreadsheet.toolPanel.show", "true");
```

### 10.3.20 spreadsheet.height

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Default Wert: 800

Legt die Höhe des spreadsheet-Dashlets fest.

```
dashlet.addOption("spreadsheet.height", "600");
```

### 10.3.21 spreadsheet.menulcon

<b>Datentyp</b>	BOOLEAN
-----------------	---------

<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.16
<b>Schlagworte / Tags</b>	

Option zum Festlegen der menuIcon

Beispiel:

```
dashlet.addOption("spreadsheet.menuIcon", "true");
```

### 10.3.22 spreadsheet.columnBorders

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Bestimmt die Anzeige der vertikalen Ränder der Spalten.

```
dashlet.addOption("spreadsheet.columnBorders", "true");
```

### 10.3.23 spreadsheet.draggable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19

### Schlagworte / Tags

Das Grid als Ganzes wird ziehbar, wenn diese Option auf true gesetzt ist.

```
dashlet.addOption("spreadsheet.draggable", "true");
```

## 10.3.24 spreadsheet.hwrap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Es bestimmt das Verhalten des Inhalts von Kopfzeilenzellen, der nicht in eine einzelne Zeile innerhalb der Breite der Zelle passt. Der Text in den Zellen wird in die nächste Zeile umgebrochen, wenn wrap = true, ansonsten wird der überlaufende Text ausgeblendet und das Fortsetzungssymbol ... wird am Ende angezeigt.

```
dashlet.addOption("spreadsheet.hwrap", "true");
```

## 10.3.25 spreadsheet.maxHeight

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Diese Option legt die maximale Höhe des Rasters in Pixel (ohne 'px'-Suffix) oder % fest.

```
dashlet.addOption("spreadsheet.maxHeight", "100");
```

### 10.3.26 spreadsheet.maxWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Diese Option legt die maximale Breite des Rasters in Pixel (ohne 'px'-Suffix) oder % fest.

```
dashlet.addOption("spreadsheet.maxWidth", "100");
```

### 10.3.27 spreadsheet.minWidth

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Minimal mögliche Breite des Rasters in Pixel.

```
dashlet.addOption("spreadsheet.minWidth", "100");
```

### 10.3.28 spreadsheet.width

<b>Datentyp</b>	STRING
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>



<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Breite des Rasters in Pixelanzahl (ohne 'px'-Suffix) zB 150, Prozent (%) zB '80%', Kombination aus % und px zB '100%-20' oder '50%+10', Auto oder flex.

```
dashlet.addOption("spreadsheet.width", "100");
```

### 10.3.29 spreadsheet.wrap

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Es bestimmt das Verhalten von Zelleninhalten, die nicht in eine einzelne Zeile innerhalb der Breite der Zelle passen. Der Text in den Zellen wird in die nächste Zeile umgebrochen, wenn wrap = true, ansonsten wird der überlaufende Text ausgeblendet und das Fortsetzungssymbol ... wird am Ende angezeigt.

```
dashlet.addOption("spreadsheet.wrap", "true");
```

### 10.3.30 spreadsheet.summaryOnTop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Standardmäßig wird die Zusammenfassung der gruppierten Zeilen oder im einfachen Raster am unteren Rand des Tabellenkörpers angezeigt. Die Zusammenfassung wird über dem Tabellenkörper angezeigt, wenn diese Option auf "true" gesetzt ist. Diese Option kann nach der Initialisierung des Rasters nicht geändert werden.

```
dashlet.addOption("spreadsheet.summaryOnTop", "true");
```

### 10.3.31 spreadsheet.roundCorners

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Anzeige von abgerundeten Ecken an allen 4 Ecken des Rasters.

```
dashlet.addOption("spreadsheet.roundCorners", "true");
```

### 10.3.32 spreadsheet.rowBorders

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Bestimmt die Anzeige der horizontalen Ränder der Zeilen.

```
dashlet.addOption("spreadsheet.rowBorders", "true");
```

### 10.3.33 spreadsheet.rowHt

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Spreadsheet Object

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Legt eine konstante Höhe aller Zeilen im Raster fest.

```
dashlet.addOption("spreadsheet.rowHt", "25");
```

### 10.3.34 spreadsheet.rowHtHead

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Legt eine konstante Höhe aller Header-Zeilen im Raster fest.

```
dashlet.addOption("spreadsheet.rowHtHead", "25");
```

### 10.3.35 spreadsheet.rowHtSum

<b>Datentyp</b>	INTEGER
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Legt eine konstante Höhe aller Footer-Zeilen im Raster fest.

```
dashlet.addOption("spreadsheet.rowHtSum ", "25");
```

### 10.3.36 spreadsheet.showBottom

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Steuert die Anzeige des unteren Abschnitts des Rasters. Eingefrorene Zeilen unten und die Paging-Symbolleiste werden im unteren Abschnitt angezeigt.

```
dashlet.addOption("spreadsheet.showBottom", "true");
```

### 10.3.37 spreadsheet.showHeader

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Er bestimmt die Anzeige der Spaltenüberschrift.

```
dashlet.addOption("spreadsheet.showHeader", "true");
```

### 10.3.38 spreadsheet.showTitle

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object

<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Steuert die Anzeige des Titels im oberen Abschnitt (über den Spaltenüberschriften) des Rasters.

```
dashlet.addOption("spreadsheet.showTitle", "true");
```

### 10.3.39 spreadsheet.showTop

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Steuert die Anzeige des oberen Abschnitts (über den Spaltenüberschriften) des Rasters.

```
dashlet.addOption("spreadsheet.showTop", "true");
```

### 10.3.40 spreadsheet.showToolbar

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Steuert die Anzeige der Symbolleiste (Toolbar) im oberen Bereich (über den Spaltenüberschriften) des Rasters.

```
dashlet.addOption("spreadsheet.showToolbar", "true");
```

### 10.3.41 spreadsheet.sortable

<b>Datentyp</b>	BOOLEAN
<b>Semantische Gruppe</b>	Spreadsheet Object
<b>Komponente</b>	Content
<b>Implementierungsstatus</b>	<b>IMPLEMENTED</b>
<b>Verfügbar seit Version</b>	2.3.19
<b>Schlagworte / Tags</b>	

Setzt der Option auf false, um die Sortierung für alle Spalten zu deaktivieren.

```
dashlet.addOption("spreadsheet.sortable", "true");
```