**Line Chart**

**Demo pic**

https://snag.gy/Qg3AuY.jpg

**Usage with url**

<appup-chart

url="https://demo7505738.mockable.io/line"

type="line"

title="linechart"

:config='{subtitle:"testing",xtitle:"this is x testing",ytitle:"this is y testing",xcategory:["first","second","third","fourth","fifth"],ycategory:["first","second","third","fourth","fifth"],tooltip:true,legend:false}'/>

**Usage with data**

<appup-chart

url="",

data=”[{"name": "Installation","values": [107, 31, 635, 203, 2]}, { "name": "Manufacturing","values": [133, 156, 947, 408, 6]}, { "name": "Sales & Distribution","values": [814, 841, 3714, 727, 31]}, { "name": "Project Development","values": [1216, 1001, 4436, 738, 40]}]”,

type="line"

title="linechart"

:config='{subtitle:"testing",xtitle:"this is x testing",ytitle:"this is y testing",xcategory:["first","second","third","fourth","fifth"],ycategory:["first","second","third","fourth","fifth"],tooltip:true,legend:false}'/>

**Different options in config:**

<appup-chart

url="https://demo7505738.mockable.io/line"

type="line"

title="linechart"

:config="{title: {

text: 'testing for line chart'

},

subtitle: {

text: 'testing'

},

yAxis: {

title: {

text: 'Number of Employees'

}

},

legend: {

layout: 'vertical',

align: 'right',

verticalAlign: 'middle'

},

plotOptions: {

series: {

label: {

connectorAllowed: false

},

pointStart: 2010

}

},

}"/>

This component is a wrapper of Highcharts. You can find documentation here

<https://www.highcharts.com/docs>

**Syntax:**

**Title:**

## align: string

The horizontal alignment of the title. Can be one of "left", "center" and "right".

Defaults to center.

## floating: boolean

When the title is floating, the plot area will not move to make space for it.

Defaults to false.

## margin: number

The margin between the title and the plot area, or if a subtitle is present, the margin between the subtitle and the plot area.

Defaults to 15.

## Style:

## CSS styles for the title. Use this for font styling, but use align, x and y for text alignment.

In **styled mode**, the title style is given in the .appupchart-title class.

Defaults to { "color": "#333333", "fontSize": "18px" }.

## text: string

The title of the chart. To disable the title, set the text to undefined.

Defaults to Chart title.

## 

## useHTML: boolean

## Whether to **use HTML** to render the text.

Defaults to false

We can use tags like <b>, <strong>, <i>, <em>, <br/>, <span>

## verticalAlign: string

The vertical alignment of the title. Can be one of "top", "middle" and "bottom". When a value is given, the title behaves as if **floating** were true.

Defaults to undefined.

## widthAdjust: number

Adjustment made to the title width, normally to reserve space for the exporting burger menu.

Defaults to -44

## x: number

The x position of the title relative to the alignment within chart.spacingLeft and chart.spacingRight.

Defaults to 0

## y: number

The y position of the title relative to the alignment within **chart.spacingTop** and **chart.spacingBottom**. By default it depends on the font size.

Defaults to undefined

Ex:

title: {

align: 'left',

floating: true,

margin: 50,

style: {

color: '#FF00FF',

fontWeight: 'bold'

},

text: 'My custom title',

Text: ‘<b>add html tags</b>’,

verticalAlign: 'bottom',

widthAdjust: -200,

x: 90,

y: 30

}

**Legend**

## align: string

The horizontal alignment of the legend box within the chart area. Valid values are left, center and right.

In the case that the legend is aligned in a corner position, the layout option will determine whether to place it above/below or on the side of the plot area.

Defaults to center

## alignColumns: boolean

If the **layout** is horizontal and the legend items span over two lines or more, whether to align the items into vertical columns. Setting this to false makes room for more items, but will look more messy.

Defaults to true.

backgroundColor:

The background color of the legend.

Defaults to undefined.

borderColor:

The color of the drawn border around the legend.

Defaults to #999999.

borderRadius: number

The border corner radius of the legend.

Defaults to 0.

borderWidth: number

The width of the drawn border around the legend.

Defaults to 0.

## enabled: boolean

Enable or disable the legend. There is also a series-specific option, showInLegend, that can hide the series from the legend. In some series types this is false by default, so it must set to true in order to show the legend for the series.

Defaults to undefined.

## floating: boolean

When the legend is floating, the plot area ignores it and is allowed to be placed below it.

Defaults to false.

## itemDistance: number

## In a legend with horizontal layout, the itemDistance defines the pixel distance between each item.

## Defaults to 20.

## itemHiddenStyle:

## CSS styles for each legend item when the corresponding series or point is hidden. Only a subset of CSS is supported, notably those options related to text. Properties are inherited from style unless overridden here.

Defaults to {"color": "#cccccc"}.

## itemHoverStyle:

## CSS styles for each legend item in hover mode. Only a subset of CSS is supported, notably those options related to text. Properties are inherited from style unless overridden here.

## Defaults to {"color": "#000000"}.

## itemMarginBottom: number

The pixel bottom margin for each legend item.

Defaults to 0.

## itemMarginTop: number

The pixel top margin for each legend item.

Defaults to 0.

## itemStyle:

## CSS styles for each legend item. Only a subset of CSS is supported, notably those options related to text. The default textOverflow property makes long texts truncate. Set it to undefined to wrap text instead. A width property can be added to control the text width.

Defaults to {"color": "#333333", "cursor": "pointer", "fontSize": "12px", "fontWeight": "bold", "textOverflow": "ellipsis"}.

## itemWidth: number

The width for each legend item. By default the items are laid out successively. In a horizontal layout, if the items are laid out across two rows or more, they will be vertically aligned depending on the legend.alignColumns option.

Defaults to undefined.

## labelFormat: string

A format string for each legend label. Available variables relates to properties on the series, or the point in case of pies.

Defaults to {name}.

## labelFormatter:

Callback function to format each of the series' labels. The this keyword refers to the series object, or the point object in case of pie charts. By default the series or point name is printed.

Defaults to undefined.

## layout: string

The layout of the legend items. Can be one of horizontal or vertical or proximate. When proximate, the legend items will be placed as close as possible to the graphs they're representing, except in inverted charts or when the legend position doesn't allow it.

Defaults to horizontal.

## margin: number

If the plot area sized is calculated automatically and the legend is not floating, the legend margin is the space between the legend and the axis labels or plot area.Defaults to 12.

## maxHeight: number

Maximum pixel height for the legend. When the maximum height is extended, navigation will show.

Defaults to undefined.

## padding: number

The inner padding of the legend box.

Defaults to 8.

## reversed: boolean

Whether to reverse the order of the legend items compared to the order of the series or points as defined in the configuration object.

Defaults to false.

rtl: boolean

Whether to show the symbol on the right side of the text rather than the left side. This is common in Arabic and Hebraic.

Defaults to false.

## shadow: boolean

## Whether to apply a drop shadow to the legend. A backgroundColor also needs to be applied for this to take effect. The shadow can be an object configuration containing color, offsetX, offsetY, opacity and width.

Defaults to false.

## squareSymbol: boolean

When this is true, the legend symbol width will be the same as the symbol height, which in turn defaults to the font size of the legend items.

Defaults to true.

## 

## symbolHeight: number

The pixel height of the symbol for series types that use a rectangle in the legend. Defaults to the font size of legend items.

Defaults to undefined.

## symbolPadding: number

The pixel padding between the legend item symbol and the legend item text.

Defaults to 5.

## symbolRadius: number

The border radius of the symbol for series types that use a rectangle in the legend. Defaults to half the symbolHeight.

Defaults to undefined.

## symbolWidth: number

The pixel width of the legend item symbol. When the squareSymbol option is set, this defaults to the symbolHeight, otherwise 16.

Defaults to undefined.

## useHTML: boolean

Whether to use HTML to render the legend item texts.

Prior to 4.1.7, when using HTML, legend.navigation was disabled.

Defaults to false.

## verticalAlign: string

The vertical alignment of the legend box. Can be one of top, middle or bottom. Vertical position can be further determined by the y option.

In the case that the legend is aligned in a corner position, the layout option will determine whether to place it above/below or on the side of the plot area.

When the layout option is proximate, the verticalAlign option doesn't apply.

Defaults to bottom.

## width: number

The width of the legend box.

Defaults to undefined.

## x: number

The x offset of the legend relative to its horizontal alignment align within chart.spacingLeft and chart.spacingRight. Negative x moves it to the left, positive x moves it to the right.

Defaults to 0.

## y: number

The vertical offset of the legend relative to it's vertical alignment verticalAlign within chart.spacingTop and chart.spacingBottom. Negative y moves it up, positive y moves it down.

Defaults to 0.

## **navigation(**It is additional element in legend**):**

Legend.navigation

Options for the paging or navigation appearing when the legend is overflown. Navigation works well on screen, but not in static exported images. One way of working around that is to **i**ncrease the chart height in export.

## activeColor:

## The color for the active up or down arrow in the legend page navigation.

Defaults to #003399.

## animation: boolean

## How to animate the pages when navigating up or down. A value of true applies the default navigation given in the chart.animation option. Additional options can be given as an object containing values for easing and duration.

Defaults to true.

## arrowSize: number

The pixel size of the up and down arrows in the legend paging navigation.

Defaults to 12.

## enabled: boolean

Whether to enable the legend navigation. In most cases, disabling the navigation results in an unwanted overflow.

See also the adapt chart to legend plugin for a solution to extend the chart height to make room for the legend, optionally in exported charts only.

Defaults to true.

## inactiveColor:

## The color of the inactive up or down arrow in the legend page navigation. .

Defaults to #cccccc.

## style:

## Text styles for the legend page navigation.

Defaults to undefined.

## **title(**It is additional element in legend**):**

Legend.title

A title to be added on top of the legend.

## style:

Generic CSS styles for the legend title.

Defaults to {"fontWeight": "bold"}.

## text: string

A text or HTML string for the title.

Defaults to undefined.

## 

## 

## 

## **keyboardNavigation(**It is additional element in legend**):**

## Keyboard navigation for the legend. Requires the Accessibility module.

## enabled: boolean

Enable/disable keyboard navigation for the legend. Requires the Accessibility module.

Defaults to true.

## 

## **itemCheckboxStyle(**It is additional element in legend**):**

legend.itemCheckboxStyle

Default styling for the checkbox next to a legend item when showCheckbox is true.

## height: string

Since 1.0.0

Defaults to 13px.

## **bubbleLegend(**It is additional element in legend**):**

The bubble legend is an additional element in legend which presents the scale of the bubble series. Individual bubble ranges can be defined by user or calculated from series. In the case of automatically calculated ranges, a 1px margin of error is permitted.

borderColor:

The color of the ranges borders, can be also defined for an individual range.

Defaults to undefined.

## borderWidth: number

The width of the ranges borders in pixels, can be also defined for an individual range.

Defaults to 2.

## className: string

An additional class name to apply to the bubble legend' circle graphical elements. This option does not replace default class names of the graphical element.

Defaults to undefined.

color:

The main color of the bubble legend. Applies to ranges, if individual color is not defined.

Defaults to undefined

## connectorClassName: string

An additional class name to apply to the bubble legend's connector graphical elements. This option does not replace default class names of the graphical element.

Defaults to undefined.

## connectorColor:

The color of the connector, can be also defined for an individual range.

Defaults to undefined.

## connectorDistance: number

The length of the connectors in pixels. If labels are centered, the distance is reduced to 0.

Defaults to 60.

## connectorWidth: number

The width of the connectors in pixels.

Defaults to 1.

## enabled: boolean

Enable or disable the bubble legend.

Defaults to false.

## legendIndex: number

The position of the bubble legend in the legend.

Defaults to 0.

## maxSize: number

Miximum bubble legend range size. If values for ranges are not specified, the minSize and the maxSize are calculated from bubble series.

Defaults to 60.

## minSize: number

Minimum bubble legend range size. If values for ranges are not specified, the minSize and the maxSize are calculated from bubble series.

Defaults to 10.

## sizeBy: string

Whether the bubble legend range value should be represented by the area or the width of the bubble. The default, area, corresponds best to the human perception of the size of each bubble.

Defaults to area.

## sizeByAbsoluteValue: boolean

When this is true, the absolute value of z determines the size of the bubble. This means that with the default zThreshold of 0, a bubble of value -1 will have the same size as a bubble of value 1, while a bubble of value 0 will have a smaller size according to minSize.

Defaults to false.

## 

## zIndex: number

Define the visual z index of the bubble legend.

Defaults to 1.

## zThreshold: number

Ranges with with lower value than zThreshold, are skipped.

Defaults to 0.

## **ranges(**It is additional element in Bubblelegend**):**

legend.bubbleLegend.ranges

Options for specific range. One range consists of bubble, label and connector.

## borderColor:

The color of the border for individual range.

Defaults to undefined.

## color:

The color of the bubble for individual range.

Defaults to undefined.

## 

## connectorColor:

## The color of the connector for individual range.

Defaults to undefined.

## value:

Range size value, similar to bubble Z data.

Defaults to undefined.

## **labels(**It is additional element in Bubblelegend**):**

legend.bubbleLegend.labels

## Options for the bubble legend labels.

## align: string

The alignment of the labels compared to the bubble legend. Can be one of left, center or right.

Defaults to right.

## allowOverlap: boolean

Whether to allow data labels to overlap.

Defaults to false.

## className: string

An additional class name to apply to the bubble legend label graphical elements. This option does not replace default class names of the graphical element.

Defaults to undefined.

## format: string

A format string for the bubble legend labels. Available variables are the same as for formatter.

Defaults to .

## Formatter:

## Available this properties are:

* this.value: The bubble value.
* this.radius: The radius of the bubble range.
* this.center: The center y position of the range.

Defaults to undefined.

## style:

CSS styles for the labels.

Defaults to undefined.

## x: number

The x position offset of the label relative to the connector.

Defaults to 0.

## y: number

The y position offset of the label relative to the connector.

Defaults to 0.

Ex:

legend: {

align: 'left',

alignColumns: true,

backgroundColor:'#FCFFC5',

borderColor: '#C98657',

borderRadius: 2,

borderWidth: 1,

enabled: true,

floating: true,

itemDistance: 50,

itemHiddenStyle: {color: ‘#cccccc’},

itemHoverStyle:{color: ‘#000000’},

itemMarginBottom: 5,

itemMarginTop: 4,

itemStyle: { lineHeight: '14px'},

itemWidth: 3,

labelFormat:’ name’,

layout: ‘horizontal’,

margin:30,

maxHeight: 7,

padding: 8,

reversed: true,

rtl: false,

shadow: true,

squareSymbol: false,

symbolHeight: 4,

symbolPadding: 5,

symbolRadius: 3,

symbolWidth: 12,

useHTML: true,

verticalAlign: ‘bottom’,

width: number,

x: 3,

y: 6,

navigation:{

activeColor: ’ #003399’,

animation:true,

arrowSize: 12,

enabled: true,

inactiveColor: '#CCC',

style: {

fontWeight: 'bold',

color: '#333',

fontSize: '12px'

}

},

title:{

style:{‘fontWeight’: ‘bold’},

text: ‘name’

},

keyboardNavigation:{

enabled: true

},

itemCheckboxStyle:{

height: ‘13px’

},

bubbleLegend:

{

borderColor: '#000000',

borderWidth: 3,

className: ‘demo’,

color: '#8bbc21',

connectorClassName: ‘name’,

connectorColor: '#000000',

connectorDistance: 40,

connectorWidth: 30,

enabled: false,

legendIndex: 3,

maxSize: 8,

minSize: 2,

sizeBy: 'width',

sizeByAbsoluteValue: true,

zIndex: 2,

zThreshold: 1,

ranges:{

borderColor: '#8bbc21',

color:'#000000',

connectorColor: ’ #003399’,

Value:,

},//ranges

labels: {

align: ‘right’,

allowOverlap: true,

className: ‘some name’,

format:'{value:.1f} mm',

style:’ #003399’,

x: 2,

y: 4

}//labels

}//bubbleLegend

}//legend

# **tooltip**

Options for the tooltip that appears when the user hovers over a series or point.

## animation: boolean

Enable or disable animation of the tooltip.

Defaults to true.

## backgroundColor:

## The background color or gradient for the tooltip.

Defaults to undefined.

## 

## borderColor:

The color of the tooltip border. When undefined, the border takes the color of the corresponding series or point.

Defaults to undefined.

## borderRadius: number

The radius of the rounded border corners.

Defaults to 3.

## borderWidth: number

The pixel width of the tooltip border.

Defaults to 1.

## enabled: boolean

Enable or disable the tooltip.

Defaults to true.

## followPointer: boolean

Whether the tooltip should follow the mouse as it moves across columns, pie slices and other point types with an extent. By default it behaves this way for scatter, bubble and pie series by override in the plotOptions for those series types.

For touch moves to behave the same way, followTouchMove must be true also.

Defaults to false.

followTouchMove: boolean

## Whether the tooltip should update as the finger moves on a touch device. If this is true and chart.panning is set,followTouchMove will take over one-finger touches, so the user needs to use two fingers for zooming and panning.

Note the difference to followPointer that only defines the *position* of the tooltip. If followPointer is false in for example a column series, the tooltip will show above or below the column, but as followTouchMove is true, the tooltip will jump from column to column as the user swipes across the plot area.

Defaults to true.

## footerFormat: string

A string to append to the tooltip format.

Defaults to .

## formatter:

## Callback function to format the text of the tooltip from scratch. Return false to disable tooltip for a specific point on series.

A subset of HTML is supported. Unless useHTML is true, the HTML of the tooltip is parsed and converted to SVG, therefore this isn't a complete HTML renderer. The following tags are supported: <b>, <strong>, <i>, <em>, <br/>, <span>. Spans can be styled with a style attribute, but only text-related CSS that is shared with SVG is handled.

Since version 2.1 the tooltip can be shared between multiple series through the shared option. The available data in the formatter differ a bit depending on whether the tooltip is shared or not. In a shared tooltip, all properties except x, which is common for all points, are kept in an array, this.points.

Available data are:

this.percentage (not shared) / this.points[i].percentage (shared)

Stacked series and pies only. The point's percentage of the total.

this.point (not shared) / this.points[i].point (shared)

The point object. The point name, if defined, is available through `this.point.name`.

this.points

In a shared tooltip, this is an array containing all other properties for each point.

this.series (not shared) / this.points[i].series (shared)

The series object. The series name is available through `this.series.name`.

this.total (not shared) / this.points[i].total (shared)

Stacked series only. The total value at this point's x value.

this.x

The x value. This property is the same regardless of the tooltip being shared or not.

this.y (not shared) / this.points[i].y (shared)

The y value.

Defaults to undefined.

## headerFormat: string

The HTML of the tooltip header line. Variables are enclosed by curly brackets. Available variables are point.key, series.name, series.color and other members from the point and series objects. The point.key variable contains the category name, x value or datetime string depending on the type of axis. For datetime axes, the point.keydate format can be set using tooltip.xDateFormat.

Defaults to <span style="font-size: 10px">{point.key}</span><br/>.

## headerShape: String

The name of a symbol to use for the border around the tooltip header. Applies only when tooltip.split is enabled.

Defaults to callout.

## hideDelay: number

The number of milliseconds to wait until the tooltip is hidden when mouse out from a point or chart.

Defaults to 500.

## outside: boolean

Whether to allow the tooltip to render outside the chart's SVG element box. By default (false), the tooltip is rendered within the chart's SVG element, which results in the tooltip being aligned inside the chart area. For small charts, this may result in clipping or overlapping. When true, a separate SVG element is created and overlaid on the page, allowing the tooltip to be aligned inside the page itself.

Defaults to false.

## padding: number

Padding inside the tooltip, in pixels.

Defaults to 8.

## pointFormat: string

The HTML of the point's line in the tooltip. Variables are enclosed by curly brackets. Available variables are point.x, point.y, series. name and series.color and other properties on the same form. Furthermore, point.y can be extended by the tooltip.valuePrefix and tooltip.valueSuffix variables. This can also be overridden for each series, which makes it a good hook for displaying units.

In styled mode, the dot is colored by a class name rather than the point color.

Defaults to <span style="color:{point.color}">\u25CF</span> {series.name}: <b>{point.y}</b><br/>.

## pointFormatter: function

A callback function for formatting the HTML output for a single point in the tooltip. Like the pointFormat string, but with more flexibility.

Defaults to undefined.

## positioner: function

A callback function to place the tooltip in a default position. The callback receives three parameters: labelWidth, labelHeight and point, where point contains values for plotX and plotY telling where the reference point is in the plot area. Add chart.plotLeft and chart.plotTop to get the full coordinates.

Since v7, when tooltip.split option is enabled, positioner is called for each of the boxes separately, including xAxis header. xAxis header is not a point, instead pointargument contains info: { plotX: Number, plotY: Number, isHeader: Boolean }

The return should be an object containing x and y values, for example { x: 100, y: 100 }.

Defaults to undefined.

shadow: boolean

Whether to apply a drop shadow to the tooltip.

Defaults to true.

## shape: string

The name of a symbol to use for the border around the tooltip. Can be one of: "callout", "circle" or "square". When tooltip.split option is enabled, shape is applied to all boxes except header, which is controlled by tooltip.headerShape.

Defaults to callout.

## shared: boolean

When the tooltip is shared, the entire plot area will capture mouse movement or touch events. Tooltip texts for series types with ordered data (not pie, scatter, flags etc) will be shown in a single bubble. This is recommended for single series charts and for tablet/mobile optimized charts.

See also tooltip.split, that is better suited for charts with many series, especially line-type series. The tooltip.split option takes precedence over tooltip.shared.

Defaults to false.

## snap: number

Proximity snap for graphs or single points. It defaults to 10 for mouse-powered devices and 25 for touch devices.

Note that in most cases the whole plot area captures the mouse movement, and in these cases tooltip.snap doesn't make sense. This applies when stickyTracking is true (default) and when the tooltip is shared or split.

Defaults to 10/25.

## split: boolean

Split the tooltip into one label per series, with the header close to the axis. This is recommended over shared tooltips for charts with multiple line series, generally making them easier to read. This option takes precedence over tooltip.shared.

Defaults to false.

## 

## 

## useHTML: boolean

Use HTML to render the contents of the tooltip instead of SVG. Using HTML allows advanced formatting like tables and images in the tooltip. It is also recommended for rtl languages as it works around rtl bugs in early Firefox.

Defaults to false.

## valueDecimals: number

How many decimals to show in each series' y value. This is overridable in each series' tooltip options object. The default is to preserve all decimals.

Defaults to undefined.

## valuePrefix: string

A string to prepend to each series' y value. Overridable in each series' tooltip options object.

Defaults to undefined.

## valueSuffix: string

A string to append to each series' y value. Overridable in each series' tooltip options object.

Defaults to undefined.

## xDateFormat: string

The format for the date in the tooltip header if the X axis is a datetime axis. The default is a best guess based on the smallest distance between points in the chart.

Defaults to undefined.

## **style(**It is additional element in tooltip**):**

CSS styles for the tooltip. The tooltip can also be styled through the CSS class .appcharts-tooltip.

## whiteSpace: string

Defaults to nowrap.

**dateTimeLabelFormats(**It is additional element in tooltip**):**

tooltip.dateTimeLabelFormats

For series on a datetime axes, the date format in the tooltip's header will by default be guessed based on the closest data points. This member gives the default string representations used for each unit. For an overview of the replacement codes, see dateFormat.

## day: string

Defaults to %A, %b %e, %Y.

## hour: string

Defaults to %A, %b %e, %H:%M.

## millisecond: string

Defaults to %A, %b %e, %H:%M:%S.%L.

## minute: string

Defaults to %A, %b %e, %H:%M.

## month: string

Defaults to %B %Y.

## second: string

Defaults to %A, %b %e, %H:%M:%S.

## week: string

Defaults to Week from %A, %b %e, %Y.

## year: string

Defaults to %Y.

Ex:

tooltip: {

animation: true',

backgroundColor: '#FCFFC5',

borderColor: '#000000',

borderRadius:3,

borderWidth:1,

enabled:true,

followPointer:false,

followTouchMove:true,

footerFormat:'</table>',

headerFormat:'<small>{point.key}</small><table>',

hideDelay:500,

outside:true,

padding:8,

pointFormat:'{series.name}: <b>{point.y}</b><br/>',

pointFormatter:function,

positioner:function,

shadow:true,

shape:’circle’,

sharped:true,

snap:50,

split:true,

style: {

color: '#FF00FF',

fontWeight: 'bold'

},

useHTML:true,

valueDecimals:2,

valuePrefix:’$’,

valueSuffix:’USD’,

xDateFormat:’%Y-%m-%d’

dateTimeLabelFormat:{

day:’%A’,

hour:’’,

millisecond:’’,

minute:’’,

month:’’,

second:’’,

week:’’,

year:’’

}

}

# **xAxis**

The X axis or category axis. Normally this is the horizontal axis, though if the chart is inverted this is the vertical axis. In case of multiple axes, the xAxis node is an array of configuration objects.

See the Axis class for programmatic access to the axis.

## alignTicks: boolean

When using multiple axis, the ticks of two or more opposite axes will automatically be aligned by adding ticks to the axis or axes with the least ticks, as if tickAmount were specified.

This can be prevented by setting alignTicks to false. If the grid lines look messy, it's a good idea to hide them for the secondary axis by setting gridLineWidth to 0.

If startOnTick or endOnTick in an Axis options are set to false, then the alignTicks will be disabled for the Axis.

Disabled for logarithmic axes.

Defaults to true.

## allowDecimals: boolean

Whether to allow decimals in this axis' ticks. When counting integers, like persons or hits on a web page, decimals should be avoided in the labels.

Defaults to true.

## 

## categories: Array.<string>

If categories are present for the xAxis, names are used instead of numbers for that axis. Since Highcharts 3.0, categories can also be extracted by giving each point a name and setting axis type to category. However, if you have multiple series, best practice remains defining the categories array.

Example:

categories: ['Apples', 'Bananas', 'Oranges']

Defaults to undefined.

## ceiling: number

Since 4.0.0

The highest allowed value for automatically computed axis extremes.

Defaults to undefined.

## className: string

A class name that opens for styling the axis by CSS, especially in Highcharts styled mode. The class name is applied to group elements for the grid, axis elements and labels.

Defaults to undefined.

## 

## alternateGridColor: Highcharts.ColorString

When using an alternate grid color, a band is painted across the plot area between every other grid line.

Defaults to undefined.

# **breaks**(It is element in xAxis):

An array defining breaks in the axis, the sections defined will be left out and all the points shifted closer to each other.

Requires that the broken-axis.js module is loaded.

# 

# **crosshair**(It is element in xAxis):

Configure a crosshair that follows either the mouse pointer or the hovered point.

In styled mode, the crosshairs are styled in the .highcharts-crosshair, .highcharts-crosshair-thin or .highcharts-xaxis-category classes.

# **Chart**

General options for the chart.

## alignTicks: boolean

When using multiple axis, the ticks of two or more opposite axes will automatically be aligned by adding ticks to the axis or axes with the least ticks, as if tickAmount were specified.

This can be prevented by setting alignTicks to false. If the grid lines look messy, it's a good idea to hide them for the secondary axis by setting gridLineWidth to 0.

If startOnTick or endOnTick in an Axis options are set to false, then the alignTicks will be disabled for the Axis.

Disabled for logarithmic axes.

Defaults to true.

## animation: boolean, Highcharts.AnimationOptionsObject

Set the overall animation for all chart updating. Animation can be disabled throughout the chart by setting it to false here. It can be overridden for each individual API method as a function parameter. The only animation not affected by this option is the initial series animation, see plotOptions.series.animation.

The animation can either be set as a boolean or a configuration object. If true, it will use the 'swing' jQuery easing and a duration of 500 ms. If used as a configuration object, the following properties are supported:

duration

The duration of the animation in milliseconds.

easing

A string reference to an easing function set on the `Math` object. See [the easing demo](https://jsfiddle.net/gh/get/library/pure/highcharts/highcharts/tree/master/samples/highcharts/plotoptions/series-animation-easing/).

Defaults to true.

## backgroundColor: Highcharts.ColorString

The background color or gradient for the outer chart area.

Defaults to #ffffff.

## borderColor: Highcharts.ColorString

The color of the outer chart border.

Defaults to #335ca

## borderRadius: number

The corner radius of the outer chart border.

Defaults to 0.

## borderWidth: number

The pixel width of the outer chart border.

Defaults to 0.

## className: string

A CSS class name to apply to the charts container div, allowing unique CSS styling for each chart.

Defaults to undefined.

## colorCount: number

In styled mode, this sets how many colors the class names should rotate between. With ten colors, series (or points) are given class names like highcharts-color-0, highcharts-color-0 [...] highcharts-color-9. The equivalent in non-styled mode is to set colors using the colors setting.

Defaults to 10.

## description: string

A text description of the chart.

If the Accessibility module is loaded, this is included by default as a long description of the chart and its contents in the hidden screen reader information region.

Defaults to undefined.

#### 

## displayErrors: boolean

Whether to display errors on the chart. When false, the errors will be shown only in the console.

Requires debugger.js module.

Defaults to true.

## height: number, string, null

An explicit height for the chart. If a *number*, the height is given in pixels. If given a *percentage string* (for example '56%'), the height is given as the percentage of the actual chart width. This allows for preserving the aspect ratio across responsive sizes.

By default (when null) the height is calculated from the offset height of the containing element, or 400 pixels if the containing element's height is 0.

## ignoreHiddenSeries: boolean

If true, the axes will scale to the remaining visible series once one series is hidden. If false, hiding and showing a series will not affect the axes or the other series. For stacks, once one series within the stack is hidden, the rest of the stack will close in around it even if the axis is not affected.

Defaults to true.

## inverted: boolean

Whether to invert the axes so that the x axis is vertical and y axis is horizontal. When true, the x axis is reversed by default.

If a bar series is present in the chart, it will be inverted automatically. Inverting the chart doesn't have an effect if there are no cartesian series in the chart, or if the chart is polar.

Defaults to false.

## margin: number, Array.<number>

The margin between the outer edge of the chart and the plot area. The numbers in the array designate top, right, bottom and left respectively. Use the options marginTop, marginRight, marginBottom and marginLeft for shorthand setting of one option.

By default there is no margin. The actual space is dynamically calculated from the offset of axis labels, axis title, title, subtitle and legend in addition to the spacingTop, spacingRight, spacingBottom and spacingLeft options.

Defaults to undefined.

## marginBottom: number

The margin between the bottom outer edge of the chart and the plot area. Use this to set a fixed pixel value for the margin as opposed to the default dynamic margin. See also spacingBottom.

Defaults to undefined.

## marginLeft: number

The margin between the left outer edge of the chart and the plot area. Use this to set a fixed pixel value for the margin as opposed to the default dynamic margin. See also spacingLeft.

Defaults to undefined.

## marginRight: number

The margin between the right outer edge of the chart and the plot area. Use this to set a fixed pixel value for the margin as opposed to the default dynamic margin. See also spacingRight.

Defaults to undefined.

#### 

## 

## marginTop: number

The margin between the top outer edge of the chart and the plot area. Use this to set a fixed pixel value for the margin as opposed to the default dynamic margin. See also spacingTop.

Defaults to undefined.

## panning: boolean

Allow panning in a chart. Best used with panKey to combine zooming and panning.

On touch devices, when the tooltip.followTouchMove option is true (default), panning requires two fingers. To allow panning with one finger, set followTouchMove to false.

Defaults to false.

## parallelCoordinates: boolean

Flag to render charts as a parallel coordinates plot. In a parallel coordinates plot (||-coords) by default all required yAxes are generated and the legend is disabled. This feature requires modules/parallel-coordinates.js.

Defaults to false.

## pinchType: string

Equivalent to zoomType, but for multitouch gestures only. By default, the pinchType is the same as the zoomType setting. However, pinching can be enabled separately in some cases, for example in stock charts where a mouse drag pans the chart, while pinching is enabled. When tooltip.followTouchMove is true, pinchType only applies to two-finger touches.

Defaults to undefined.

## plotBackgroundColor: Highcharts.ColorString

The background color or gradient for the plot area.

Defaults to undefined.

## 

## 

## plotBackgroundImage: string

The URL for an image to use as the plot background. To set an image as the background for the entire chart, set a CSS background image to the container element. Note that for the image to be applied to exported charts, its URL needs to be accessible by the export server.

Defaults to undefined.

## plotBorderColor: Highcharts.ColorString

The color of the inner chart or plot area border.

Defaults to #cccccc.

## plotBorderWidth: number

The pixel width of the plot area border.

Defaults to 0.

## plotShadow: boolean, Highcharts.CSSObject

Whether to apply a drop shadow to the plot area. Requires that plotBackgroundColor be set. The shadow can be an object configuration containing color, offsetX, offsetY, opacity and width.

Defaults to false.

## polar: boolean

When true, cartesian charts like line, spline, area and column are transformed into the polar coordinate system. This produces *polar charts*, also known as *radar charts*. Requires highcharts-more.js.

Defaults to false.

## reflow: boolean

Whether to reflow the chart to fit the width of the container div on resizing the window.

Defaults to true.

## renderTo: string, Highcharts.SVGDOMElement

The HTML element where the chart will be rendered. If it is a string, the element by that id is used. The HTML element can also be passed by direct reference, or as the first argument of the chart constructor, in which case the option is not needed.

Defaults to undefined.

## selectionMarkerFill: Highcharts.ColorString

The background color of the marker square when selecting (zooming in on) an area of the chart.

Defaults to rgba(51,92,173,0.25).

## 

## shadow: boolean, Highcharts.CSSObject

Whether to apply a drop shadow to the outer chart area. Requires that backgroundColor be set. The shadow can be an object configuration containing color, offsetX, offsetY, opacity and width.

Defaults to false.

## showAxes: boolean

Whether to show the axes initially. This only applies to empty charts where series are added dynamically, as axes are automatically added to cartesian series.

Defaults to undefined.

## spacing: Array.<number>

The distance between the outer edge of the chart and the content, like title or legend, or axis title and labels if present. The numbers in the array designate top, right, bottom and left respectively. Use the options spacingTop, spacingRight, spacingBottom and spacingLeft options for shorthand setting of one option.

Defaults to [10, 10, 15, 10].

## spacingBottom: number

The space between the bottom edge of the chart and the content (plot area, axis title and labels, title, subtitle or legend in top position).

Defaults to 15.

## spacingLeft: number

The space between the left edge of the chart and the content (plot area, axis title and labels, title, subtitle or legend in top position).

Defaults to 10.

## spacingRight: number

The space between the right edge of the chart and the content (plot area, axis title and labels, title, subtitle or legend in top position).

Defaults to 10.

## spacingTop: number

The space between the top edge of the chart and the content (plot area, axis title and labels, title, subtitle or legend in top position).

Defaults to 10.

## 

## style: Highcharts.CSSObject

Additional CSS styles to apply inline to the container div. Note that since the default font styles are applied in the renderer, it is ignorant of the individual chart options and must be set globally.

Defaults to {"fontFamily": "\"Lucida Grande\", \"Lucida Sans Unicode\", Verdana, Arial, Helvetica, sans-serif","fontSize":"12px"}.

## styledMode: boolean

Whether to apply styled mode. When in styled mode, no presentational attributes or CSS are applied to the chart SVG. Instead, CSS rules are required to style the chart. The default style sheet is available from https://code.highcharts.com/css/highcharts.css.

Defaults to false.

## type: string

The default series type for the chart. Can be any of the chart types listed under plotOptions.

Defaults to line.

## typeDescription: string

A text description of the chart type.

If the Accessibility module is loaded, this will be included in the description of the chart in the screen reader information region.

Highcharts will by default attempt to guess the chart type, but for more complex charts it is recommended to specify this property for clarity.

Defaults to undefined.

## width: number, null

An explicit width for the chart. By default (when null) the width is calculated from the offset width of the containing element.

Defaults to null.

## zoomKey: string

Set a key to hold when dragging to zoom the chart. Requires the draggable-points module. This is useful to avoid zooming while moving points. Should be set different than chart.panKey.

Defaults to undefined.

## zoomType: string

Decides in what dimensions the user can zoom by dragging the mouse. Can be one of x, y or xy.

Defaults to undefined.

## 

## 

## **parallelAxes(**It is element in chart**):**

## chart.parallelAxes

Common options for all yAxes rendered in a parallel coordinates plot. This feature requires modules/parallel-coordinates.js.

The default options are:

parallelAxes: {  
 lineWidth: 1, // classic mode only  
 gridlinesWidth: 0, // classic mode only  
 title: {  
 text: '',  
 reserveSpace: false  
 },  
 labels: {  
 x: 0,  
 y: 0,  
 align: 'center',  
 reserveSpace: false  
 },  
 offset: 0  
}

## alignTicks: boolean

When using multiple axis, the ticks of two or more opposite axes will automatically be aligned by adding ticks to the axis or axes with the least ticks, as if tickAmount were specified.

This can be prevented by setting alignTicks to false. If the grid lines look messy, it's a good idea to hide them for the secondary axis by setting gridLineWidth to 0.

If startOnTick or endOnTick in an Axis options are set to false, then the alignTicks will be disabled for the Axis.

Disabled for logarithmic axes.

Defaults to true.

## allowDecimals: boolean

Whether to allow decimals in this axis' ticks. When counting integers, like persons or hits on a web page, decimals should be avoided in the labels.

Defaults to true.

## categories: Array.<string>

If categories are present for the xAxis, names are used instead of numbers for that axis. Since Highcharts 3.0, categories can also be extracted by giving each point a name and setting axis type to category. However, if you have multiple series, best practice remains defining the categories array.

Example:

categories: ['Apples', 'Bananas', 'Oranges']

Defaults to undefined.

## ceiling: number

The highest allowed value for automatically computed axis extremes.

Defaults to undefined.

## className: string

A class name that opens for styling the axis by CSS, especially in Highcharts styled mode. The class name is applied to group elements for the grid, axis elements and labels.

Defaults to undefined

## description: string

Requires Accessibility module

Description of the axis to screen reader users.

Defaults to undefined.

## endOnTick: boolean

Whether to force the axis to end on a tick. Use this option with the maxPadding option to control the axis end.

Defaults to true.

## floor: number

The lowest allowed value for automatically computed axis extremes.

Defaults to undefined.

## gridZIndex: number

The Z index of the grid lines.

Defaults to 1.

## lineColor: Highcharts.ColorString

The color of the line marking the axis itself.

In styled mode, the line stroke is given in the .highcharts-axis-line or .highcharts-xaxis-line class.

Defaults to #ccd6eb.

## lineWidth: number

The width of the line marking the axis itself.

In styled mode, the stroke width is given in the .highcharts-axis-line or .highcharts-xaxis-line class.

Defaults to 1.

## linkedTo: number

Index of another axis that this axis is linked to. When an axis is linked to a master axis, it will take the same extremes as the master, but as assigned by min or max or by setExtremes. It can be used to show additional info, or to ease reading the chart by duplicating the scales.

Defaults to undefined.

## max: number

The maximum value of the axis. If null, the max value is automatically calculated.

If the endOnTick option is true, the max value might be rounded up.

If a tickAmount is set, the axis may be extended beyond the set max in order to reach the given number of ticks. The same may happen in a chart with multiple axes, determined by chart. alignTicks, where a tickAmount is applied internally.

Defaults to undefined.

## maxPadding: number

Padding of the max value relative to the length of the axis. A padding of 0.05 will make a 100px axis 5px longer. This is useful when you don't want the highest data value to appear on the edge of the plot area. When the axis' max option is set or a max extreme is set using axis.setExtremes(), the maxPadding will be ignored.

Defaults to 0.05.

## min: number

The minimum value of the axis. If null the min value is automatically calculated.

If the startOnTick option is true (default), the min value might be rounded down.

The automatically calculated minimum value is also affected by floor, softMin, minPadding, minRange as well as series.threshold and series.softThreshold.

Defaults to undefined.

## minorTickColor: Highcharts.ColorString

Color for the minhttps://api.highcharts.com/highcharts/tooltip.followTouchMove  
or tick marks.

Defaults to #999999.

## minorTickInterval: number, string, null

Specific tick interval in axis units for the minor ticks. On a linear axis, if "auto", the minor tick interval is calculated as a fifth of the tickInterval. If null or undefined, minor ticks are not shown.

On logarithmic axes, the unit is the power of the value. For example, setting the minorTickInterval to 1 puts one tick on each of 0.1, 1, 10, 100 etc. Setting the minorTickInterval to 0.1 produces 9 ticks between 1 and 10, 10 and 100 etc.

If user settings dictate minor ticks to become too dense, they don't make sense, and will be ignored to prevent performance problems.

Defaults to undefined.

## minorTickLength: number

The pixel length of the minor tick marks.

Defaults to 2.

minorTickPosition: string

The position of the minor tick marks relative to the axis line. Can be one of inside and outside.

Defaults to outside.

## 

## minorTicks: boolean

Enable or disable minor ticks. Unless minorTickInterval is set, the tick interval is calculated as a fifth of the tickInterval.

On a logarithmic axis, minor ticks are laid out based on a best guess, attempting to enter approximately 5 minor ticks between each major tick.

Prior to v6.0.0, ticks were unabled in auto layout by setting minorTickInterval to "auto".

On axes using categories, minor ticks are not supported.

Defaults to false.

## minorTickWidth: number

The pixel width of the minor tick mark.

Defaults to 0.

## minPadding: number

Padding of the min value relative to the length of the axis. A padding of 0.05 will make a 100px axis 5px longer. This is useful when you don't want the lowest data value to appear on the edge of the plot area. When the axis' min option is set or a max extreme is set using axis.setExtremes(), the maxPadding will be ignored.

Defaults to 0.05.

## minRange: number

The minimum range to display on this axis. The entire axis will not be allowed to span over a smaller interval than this. For example, for a datetime axis the main unit is milliseconds. If minRange is set to 3600000, you can't zoom in more than to one hour.

The default minRange for the x axis is five times the smallest interval between any of the data points.

On a logarithmic axis, the unit for the minimum range is the power. So a minRange of 1 means that the axis can be zoomed to 10-100, 100-1000, 1000-10000 etc.

Note that the minPadding, maxPadding, startOnTick and endOnTick settings also affect how the extremes of the axis are computed.

Defaults to undefined.

## minTickInterval: number

The minimum tick interval allowed in axis values. For example on zooming in on an axis with daily data, this can be used to prevent the axis from showing hours. Defaults to the closest distance between two points on the axis.

Defaults to undefined.

## offset: number

The distance in pixels from the plot area to the axis line. A positive offset moves the axis with it's line, labels and ticks away from the plot area. This is typically used when two or more axes are displayed on the same side of the plot. With multiple axes the offset is dynamically adjusted to avoid collision, this can be overridden by setting offset explicitly.Defaults to 0.

opposite: boolean

Whether to display the axis on the opposite side of the normal. The normal is on the left side for vertical axes and bottom for horizontal, so the opposite sides will be right and top respectively. This is typically used with dual or multiple axes.

Defaults to false.

## pane: number

Refers to the index in the panes array. Used for circular gauges and polar charts. When the option is not set then first pane will be used.

Defaults to undefined.

## reversed: boolean

Whether to reverse the axis so that the highest number is closest to the origin.

Defaults to false.

## reversedStacks: boolean

If true, the first series in a stack will be drawn on top in a positive, non-reversed Y axis. If false, the first series is in the base of the stack.

Defaults to true.

## showEmpty: boolean

Whether to show the axis line and title when the axis has no data.

Defaults to true.

## showFirstLabel: boolean

Whether to show the first tick label.

Defaults to true.

## showLastLabel: boolean

Whether to show the last tick label. Defaults to true on cartesian charts, and false on polar charts.

Defaults to true.

## softMax: number

A soft maximum for the axis. If the series data maximum is less than this, the axis will stay at this maximum, but if the series data maximum is higher, the axis will flex to show all data.

**Note**: The series.softThreshold option takes precedence over this option.

Defaults to undefined.

## 

## softMin: number

A soft minimum for the axis. If the series data minimum is greater than this, the axis will stay at this minimum, but if the series data minimum is lower, the axis will flex to show all data.

**Note**: The series.softThreshold option takes precedence over this option.

Defaults to undefined.

## startOfWeek: number

For datetime axes, this decides where to put the tick between weeks. 0 = Sunday, 1 = Monday.

Defaults to 1.

## startOnTick: boolean

Whether to force the axis to start on a tick. Use this option with the maxPadding option to control the axis start.

Defaults to true.

## tickAmount: number

The amount of ticks to draw on the axis. This opens up for aligning the ticks of multiple charts or panes within a chart. This option overrides the tickPixelInterval option.

This option only has an effect on linear axes. Datetime, logarithmic or category axes are not affected.

Defaults to undefined.

## tickColor: Highcharts.ColorString

Color for the main tick marks.

In styled mode, the stroke is given in the .highcharts-tick class.

Defaults to #ccd6eb.

## tickInterval: number

The interval of the tick marks in axis units. When undefined, the tick interval is computed to approximately follow the tickPixelInterval on linear and datetime axes. On categorized axes, a undefined tickInterval will default to 1, one category. Note that datetime axes are based on milliseconds, so for example an interval of one day is expressed as 24 \* 3600 \* 1000.

On logarithmic axes, the tickInterval is based on powers, so a tickInterval of 1 means one tick on each of 0.1, 1, 10, 100 etc. A tickInterval of 2 means a tick of 0.1, 10, 1000 etc. A tickInterval of 0.2 puts a tick on 0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 6, 8, 10, 20, 40 etc.

If the tickInterval is too dense for labels to be drawn, Highcharts may remove ticks.

If the chart has multiple axes, the alignTicks option may interfere with the tickInterval setting.

Defaults to undefined.

## tickLength: number

The pixel length of the main tick marks.

Defaults to 10.

tickmarkPlacement: string

For categorized axes only. If on the tick mark is placed in the center of the category, if between the tick mark is placed between categories. The default is between if the tickInterval is 1, else on.

Defaults to between.

## tickPixelInterval: number

If tickInterval is null this option sets the approximate pixel interval of the tick marks. Not applicable to categorized axis.

The tick interval is also influenced by the minTickInterval option, that, by default prevents ticks from being denser than the data points.

Defaults to 72.

## tickPosition: string

The position of the major tick marks relative to the axis line. Can be one of inside and outside.

Defaults to outside.

## tickPositioner: Highcharts.AxisTickPositionerCallbackFunction

A callback function returning array defining where the ticks are laid out on the axis. This overrides the default behaviour of tickPixelInterval and tickInterval. The automatic tick positions are accessible through this.tickPositions and can be modified by the callback.

Defaults to undefined.

## tickPositions: Array.<number>

An array defining where the ticks are laid out on the axis. This overrides the default behaviour of tickPixelInterval and tickInterval.

Defaults to undefined.

## tickWidth: number

The pixel width of the major tick marks.

Defaults to 0.

## tooltipValueFormat: string

Parallel coordinates only. Format that will be used for point.y and available in tooltip.pointFormat as {point.formattedValue}. If not set, {point.formattedValue} will use other options, in this order:

1. yAxis.labels.format will be used if set
2. If yAxis is a category, then category name will be displayed
3. If yAxis is a datetime, then value will use the same format as yAxis labels
4. If yAxis is linear/logarithmic type, then simple value will be used
5. Defaults to undefined.

## type: string

The type of axis. Can be one of linear, logarithmic, datetime, category or treegrid. Defaults to treegrid for Gantt charts, linear for other chart types.

In a datetime axis, the numbers are given in milliseconds, and tick marks are placed on appropriate values, like full hours or days. In a category or treegrid axis, the point names of the chart's series are used for categories, if a categories array is not defined.

Defaults to linear.

## uniqueNames: boolean

Applies only when the axis type is category. When uniqueNames is true, points are placed on the X axis according to their names. If the same point name is repeated in the same or another series, the point is placed on the same X position as other points of the same name. When uniqueNames is false, the points are laid out in increasing X positions regardless of their names, and the X axis category will take the name of the last point in each position.

Defaults to true.

## units: Array.<Array.<string, (Array.<number>|null)>>

Datetime axis only. An array determining what time intervals the ticks are allowed to fall on. Each array item is an array where the first value is the time unit and the second value another array of allowed multiples. Defaults to:

units: [[  
 'millisecond', // unit name  
 [1, 2, 5, 10, 20, 25, 50, 100, 200, 500] // allowed multiples  
], [  
 'second',  
 [1, 2, 5, 10, 15, 30]  
], [  
 'minute',  
 [1, 2, 5, 10, 15, 30]  
], [  
 'hour',  
 [1, 2, 3, 4, 6, 8, 12]  
], [  
 'day',  
 [1]  
], [  
 'week',  
 [1]  
], [  
 'month',  
 [1, 3, 6]  
], [  
 'year',  
 null  
]]

Defaults to undefined.

## visible: boolean

Whether axis, including axis title, line, ticks and labels, should be visible.

Defaults to true.

## **crosshair(**It is element in paralleAxis**):**

Configure a crosshair that follows either the mouse pointer or the hovered point.

In styled mode, the crosshairs are styled in the .highcharts-crosshair, .highcharts-crosshair-thin or highcharts-xaxis-category classes.

## className: string

A class name for the crosshair, especially as a hook for styling.

Defaults to undefined.

## color: Highcharts.ColorString

The color of the crosshair. Defaults to #cccccc for numeric and datetime axes, and rgba(204,214,235,0.25) for category axes, where the crosshair by default highlights the whole category.

Defaults to #cccccc.

## 

## dashStyle: string

The dash style for the crosshair. See series.dashStyle for possible values.

Defaults to Solid.

## snap: boolean

Whether the crosshair should snap to the point or follow the pointer independent of points.

Defaults to true.

## width: number

The pixel width of the crosshair. Defaults to 1 for numeric or datetime axes, and for one category width for category axes.

Defaults to 1.

## zIndex: number

The Z index of the crosshair. Higher Z indices allow drawing the crosshair on top of the series or behind the grid lines.

Defaults to 2.

## **dateTimeLabelFormats(**It is element in paralleAxis**):**

For a datetime axis, the scale will automatically adjust to the appropriate unit. This member gives the default string representations used for each unit. For intermediate values, different units may be used, for example the day unit can be used on midnight and hour unit be used for intermediate values on the same axis. For an overview of the replacement codes, see dateFormat. Defaults to:

{  
 millisecond: '%H:%M:%S.%L',  
 second: '%H:%M:%S',  
 minute: '%H:%M',  
 hour: '%H:%M',  
 day: '%e. %b',  
 week: '%e. %b',  
 month: '%b \'%y',  
 year: '%Y'  
}

## **day(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %e. %b.

## **hour(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %H:%M.

## range: boolean

Defaults to false.

## **millisecond(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %H:%M:%S.%L.

## range: boolean

Defaults to false.

## **minute(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %H:%M.

## range: boolean

Defaults to false.

## **month(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %b ‘%y.

## **second(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %H:%M:%S.

## range: boolean

Defaults to false.

## **week(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %e. %b.

## **year(**It is element in dateTimeLabelFormats**):**

## main: string

Defaults to %Y.

## **events(**It is element in paralleAxis**):**

# chart.parallelAxes.events

Event handlers for the axis.

## 

## afterBreaks: Highcharts.AxisEventCallbackFunction

An event fired after the breaks have rendered.

Defaults to undefined.

## afterSetExtremes: Highcharts.AxisEventCallbackFunction

As opposed to the setExtremes event, this event fires after the final min and max values are computed and corrected for minRange.

Fires when the minimum and maximum is set for the axis, either by calling the .setExtremes() method or by selecting an area in the chart. One parameter, event, is passed to the function, containing common event information.

The new user set minimum and maximum values can be found by event.min and event.max. These reflect the axis minimum and maximum in axis values. The actual data extremes are found in event.dataMin and event.dataMax.

Defaults to undefined.

## pointBreak: Highcharts.AxisPointBreakEventCallbackFunction

An event fired when a break from this axis occurs on a point.

Defaults to undefined.

## pointInBreak: Highcharts.AxisPointBreakEventCallbackFunction

An event fired when a point falls inside a break from this axis.

Defaults to undefined.

## setExtremes: Highcharts.AxisSetExtremesEventCallbackFunction

Fires when the minimum and maximum is set for the axis, either by calling the .setExtremes() method or by selecting an area in the chart. One parameter, event, is passed to the function, containing common event information.

The new user set minimum and maximum values can be found by event.min and event.max. These reflect the axis minimum and maximum in data values. When an axis is zoomed all the way out from the "Reset zoom" button, event.min and event.max are null, and the new extremes are set based on this.dataMin and this.dataMax.

Defaults to undefined.

## **labels(**It is element in paralleAxis**):**

# chart.parallelAxes.labels

The axis labels show the number or category for each tick.

## align: string

What part of the string the given position is anchored to. Can be one of "left", "center" or "right". The exact position also depends on the labels.x setting.

Angular gauges and solid gauges defaults to center.

Defaults to center.

## autoRotation: Array.<number>

For horizontal axes, the allowed degrees of label rotation to prevent overlapping labels. If there is enough space, labels are not rotated. As the chart gets narrower, it will start rotating the labels -45 degrees, then remove every second label and try again with rotations 0 and -45 etc. Set it to false to disable rotation, which will cause the labels to word-wrap if possible.

Defaults to [-45].

## autoRotationLimit: number

When each category width is more than this many pixels, we don't apply auto rotation. Instead, we lay out the axis label with word wrap. A lower limit makes sense when the label contains multiple short words that don't extend the available horizontal space for each label.

Defaults to 80.

## distance: number

Angular gauges and solid gauges only. The label's pixel distance from the perimeter of the plot area.

Defaults to -25.

## enabled: boolean

Enable or disable the axis labels.

Defaults to true.

## format: string

A format string for the axis label.

Defaults to {value}.

## formatter: Highcharts.FormatterCallbackFunction.<Highcharts.AxisLabelsFormatterContextObject>

Callback JavaScript function to format the label. The value is given by this.value. Additional properties for this are axis, chart, isFirst and isLast. The value of the default label formatter can be retrieved by calling this.axis.defaultLabelFormatter.call(this) within the function.

Defaults to:

function() {  
 return this.value;  
}

Defaults to undefined.

## maxStaggerLines: number

Horizontal axis only. When staggerLines is not set, maxStaggerLines defines how many lines the axis is allowed to add to automatically avoid overlapping X labels. Set to 1 to disable overlap detection.

Defaults to 5.

## overflow: boolean, string

How to handle overflowing labels on horizontal axis. If set to "allow", it will not be aligned at all. By default it "justify" labels inside the chart area. If there is room to move it, it will be aligned to the edge, else it will be removed.

Defaults to justify.

## padding: number

The pixel padding for axis labels, to ensure white space between them.

Defaults to 5.

## position3d: string

Defines how the labels are be repositioned according to the 3D chart orientation.

* 'offset': Maintain a fixed horizontal/vertical distance from the tick marks, despite the chart orientation. This is the backwards compatible behavior, and causes skewing of X and Z axes.
* 'chart': Preserve 3D position relative to the chart. This looks nice, but hard to read if the text isn't forward-facing.
* 'flap': Rotated text along the axis to compensate for the chart orientation. This tries to maintain text as legible as possible on all orientations.
* 'ortho': Rotated text along the axis direction so that the labels are orthogonal to the axis. This is very similar to 'flap', but prevents skewing the labels (X and Y scaling are still present).

Defaults to offset.

## reserveSpace: boolean

Whether to reserve space for the labels. By default, space is reserved for the labels in these cases:

* On all horizontal axes.
* On vertical axes if label.align is right on a left-side axis or left on a right-side axis.
* On vertical axes if label.align is center.

This can be turned off when for example the labels are rendered inside the plot area instead of outside.

Defaults to false.

## rotation: number

Rotation of the labels in degrees.

Defaults to 0.

## skew3d: boolean

If enabled, the axis labels will skewed to follow the perspective.

This will fix overlapping labels and titles, but texts become less legible due to the distortion.

The final appearance depends heavily on labels.position3d.

Defaults to false.

## staggerLines: number

Horizontal axes only. The number of lines to spread the labels over to make room or tighter labels.

Defaults to undefined.

## step: number

To show only every *n\_'th label on the axis, set the step to \_n*. Setting the step to 2 shows every other label.

By default, the step is calculated automatically to avoid overlap. To prevent this, set it to 1. This usually only happens on a category axis, and is often a sign that you have chosen the wrong axis type.

Read more at Axis docs => What axis should I use?

Defaults to undefined.

## style: Highcharts.CSSObject

CSS styles for the label. Use whiteSpace: 'nowrap' to prevent wrapping of category labels. Use textOverflow: 'none' to prevent ellipsis (dots).

In styled mode, the labels are styled with the .highcharts-axis-labels class.

Defaults to {"color": "#666666", "cursor": "default", "fontSize": "11px"}.

## useHTML: boolean

Whether to use HTML to render the labels.

Defaults to false.

## x: number

The x position offset of the label relative to the tick position on the axis. Defaults to -15 for left axis, 15 for right axis.

Defaults to 0.

## y: number

The y position offset of the label relative to the tick position on the axis.

Defaults to 4.

zIndex: number

The Z index for the axis labels.

Defaults to 7.

## **title(**It is element in paralleAxis**):**

# chart.parallelAxes.title

Titles for yAxes are taken from xAxis.categories. All options for xAxis.labels applies to parallel coordinates titles. For example, to style categories, use xAxis.labels.style.

## textAlign: string

Alignment of the text, can be "left", "right" or "center". Default alignment depends on the title.align:

Horizontal axes:

* for align = "low", textAlign is set to left
* for align = "middle", textAlign is set to center
* for align = "high", textAlign is set to right

Vertical axes:

* for align = "low" and opposite = true, textAlign is set to right
* for align = "low" and opposite = false, textAlign is set to left
* for align = "middle", textAlign is set to center
* for align = "high" and opposite = true textAlign is set to left
* for align = "high" and opposite = false textAlign is set to right

Defaults to undefined.

## 

## 

## **resetZoomButton(**It is element in chart**):**

The button that appears after a selection zoom, allowing the user to reset zoom.

## relativeTo: string

What frame the button should be placed related to. Can be either plot or chart

Defaults to plot.

## **position(**It is element in resetZoomButton**):**

# chart.resetZoomButton.position

The position of the button.

## align: string

The horizontal alignment of the button.

Defaults to right.

## verticalAlign: string

The vertical alignment of the button.

Defaults to top.

## x: number

The horizontal offset of the button.

Defaults to -10.

## y: number

The vertical offset of the button.

Defaults to 10.

## 

## 

## **theme(**It is element in resetZoomButton**):**

# chart.resetZoomButton.theme

A collection of attributes for the button. The object takes SVG attributes like fill, stroke, stroke-width or r, the border radius. The theme also supports style, a collection of CSS properties for the text. Equivalent attributes for the hover state are given in theme.states.hover.

## zIndex: number

The Z index for the reset zoom button. The default value places it below the tooltip that has Z index 7.

Defaults to 6.

## **scrollablePlotArea(**It is element in chart**):**

Options for a scrollable plot area. This feature provides a minimum width for the plot area of the chart. If the width gets smaller than this, typically on mobile devices, a native browser scrollbar is presented below the chart. This scrollbar provides smooth scrolling for the contents of the plot area, whereas the title, legend and axes are fixed.

## minWidth: number

The minimum width for the plot area. If it gets smaller than this, the plot area will become scrollable.

Defaults to undefined.

## scrollPositionX: number

The initial scrolling position of the scrollable plot area. Ranges from 0 to 1, where 0 aligns the plot area to the left and 1 aligns it to the right. Typically we would use 1 if the chart has right aligned Y axes.

Defaults to undefined.

Ex.

chart: {

alignTicks:true

## Animation:true

## backgroundColor:#ffffff

## borderColor: 'blue',

## borderWidth: 2,

## 

## borderRadius: 500

## 

## className:undefined

## colorCount:10

## Description:undefined

## displayErrors:true

height:40

ignoreHiddenSeries:true

Inverted:false

margin: [40, 50, 50, 50]

marginBottom:100

marginLeft:200

marginRight:125

marginTop:120

Panning:true

parallelCoordinates: true,(not working appup)

pinchType:undefined

plotBackgroundColor: '#FCFFC5'

plotBackgroundImage: '<https://www.highcharts.com/samples/graphics/skies.jpg>'

plotBorderColor: '#346691',

plotBorderWidth: 8

plotShadow:true

polar: true,

reflow:false

renderTo: container,

selectionMarkerFill:rgba(51,92,173,0.25)

Shadow:false

showAxes: true

spacing:[10, 10, 15, 10]

spacingBottom:15

spacingLeft:10

spacingRight:10

spacingTop:10

style: {

fontFamily: 'serif'

}

styledMode:false

type:’bar’

typeDescription:undefined

width:200

zoomKey:undefined

zoomType: 'x'

parallelAxes:{

alignTicks:true

allowDecimals:true//it is working on axises

categories:['Jan', 'Feb', 'Mar', 'Apr', 'May']

ceiling: 100,

className: 'highcharts-color-1',

description:undefined

endOnTick: false

floor: 30,

gridZIndex:1

lineColor: '#FF0000',

lineWidth: 10

linkedTo:{

linkedTo: 0,

opposite: true

}

max: 500

maxPadding:0.05

min: -50

minorTickColor:#999999

minorTickInterval:0.1

minorTickLength:2

minorTickPosition:outside

minorTicks:false

minorTickWidth:2

minPadding:0.05

minRange:undefined

minTickInterval:2

offset:0

opposite:false

pane:undefined

reversed:false

reversedStacks:true

showEmpty:true

showFirstLabel:true

showLastLabel:true

softMax:100

softMin:100

startOfWeek:1

startOnTick:true

tickAmount:2

tickColor:#ccd6eb

tickInterval: 5

tickLength: 100

tickmarkPlacement:between

tickPixelInterval:72

tickPosition:outside

tickPositioner: function () {

var positions = [],

tick = Math.floor(this.dataMin),

increment = Math.ceil((this.dataMax - this.dataMin) / 6);

if (this.dataMax !== null && this.dataMin !== null) {

for (tick; tick - increment <= this.dataMax; tick += increment) {

positions.push(tick);

}

}

return positions;

}

tickPositions: [0, 1, 2, 4, 8]

tickWidth: 10

tooltip: {

pointFormat: '<span style="color:{point.color}">\u25CF</span> {series.name}: <b>{point.formattedValue}</b><br/>'

},

type:linear

uniqueNames:true

units:undefined

visible:true

Crosshair:{

className:undefined

color:#cccccc

dashStyle:Solid

snap:true

width:1

zIndex:2

}

dateTimeLabelFormats:{

day:{}

hour:{}

millisecond:{}

minute:{}

month:{}

second:{}

week:{}

year:{}

}

Events:{

afterBreaks:undefined

afterSetExtremes:undefined

pointBreak:undefined

pointInBreak:undefined

setExtremes:undefined

}

labels:{

align:center

autoRotation:[-45]

autoRotationLimit:80

distance:-25

enabled:true

format:{value}

formatter:undefined

maxStaggerLines:5

overflow:justify

padding:5

position3d:offset

reserveSpace:false

rotation:0

skew3d:false

staggerLines:undefined

step:undefined

style:{"color": "#666666", "cursor": "default", "fontSize": "11px"}

useHTML:false

x[:](https://api.highcharts.com/highcharts/chart.parallelAxes.labels.x)0

y:4

zIndex:7

}

title:{

textAlign: string

}

}

resetZoomButton:{

position:{

align:right

verticalAlign:top

x:-10

y:10

}

relativeTo:plot

theme:{

zIndex:6

}

}

scrollablePlotArea:{

minWidth:undefined

scrollPositionX:undefined

}

# 