



Inhaltsverzeichnis

1. Vorlage:Graph:Chart	2
2. Vorlage:Graph:Chart/styles.css	6

Vorlage:Graph:Chart

Vorlage:Lua Vorlage:Uses TemplateStyles

Parameters

chart

Creates a JSON object for <graph> to display charts. In the article namespace the template **Vorlage:TI** should be used instead. See its page for use cases.

Parameters:

- **width:** width of the chart
- **height:** height of the chart
- **type:** type of the chart: `line` for **line charts**, `area` for **area charts**, and `rect` for (column) **bar charts**, and `pie` for **pie charts**. Multiple series can stacked using the stacked prefix, e.g. `stackedarea`.
- **interpolate:** **interpolation** method for line and area charts. It is recommended to use `monotone` for a **monotone cubic interpolation** – further supported values are listed at <https://github.com/vega/vega/wiki/Marks#area>.
- **colors:** color palette of the chart as a comma-separated list of colors. The color values must be given either as `#rgb/#rrggbb/#aarrggbb` or by a **CSS color name**. For `#aarrggbb` the `aa` component denotes the **alpha channel**, i.e. `FF`=100% opacity, `80`=50% opacity/transparency, etc. (The default color palette is `category10`).
- **xAxisTitle** and **yAxisTitle:** captions of the x and y axes
- **xAxisMin**, **xAxisMax**, **yAxisMin**, and **yAxisMax:** minimum and maximum values of the x and y axes
- **xAxisFormat** and **yAxisFormat:** changes the formatting of the axis labels. Supported values are listed at <https://github.com/d3/d3-3.x-api-reference/blob/master/Formatting.md#numbers> for numbers and <https://github.com/d3/d3-3.x-api-reference/blob/master/Time-Formatting.md> for date/time. For example, the format `%` can be used to output percentages.
- **xAxisAngle:** rotates the x axis labels by the specified angle. Recommended values are: `-45`, `+45`, `-90`, `+90`
- **xType** and **yType:** Data types of the values, e.g. `integer` for integers, `number` for real numbers, `date` for dates (e.g. `YYYY/MM/DD`), and `string` for ordinal values.
- **x:** the x-values as a comma-separated list (if a value itself contains a comma it must be escaped with a backslash, i.e. it needs to be written as `\,`)
- **y** or **y1**, **y2**, ...: the y-values for one or several data series, respectively. For pie charts `y2` denotes the radiuses of the corresponding sectors.
- **legend:** show legend (only works in case of multiple data series)
- **y1Title**, **y2Title**, ...: defines the label of the respective data series in the legend
- **linewidth:** line width for line charts or distance between the pie segments for pie charts



- **showValues:** Additionally, output the y values as text. (Currently, only (non-stacked) bar and pie charts are supported.) The output can be configured used the following parameters provided as name1:value1, name2:value2:
 - **format:** Format the output according to <https://github.com/d3/d3-3.x-api-reference/blob/master/Formatting.md#numbers> for numbers and <https://github.com/d3/d3-3.x-api-reference/blob/master/Time-Formatting.md> for date/time.
 - **fontcolor:** text color
 - **fontsize:** text size
 - **offset:** move text by the given offset. For bar charts and pie charts with `midangle` this also defines if the text is inside or outside the chart.
 - **angle** (pie charts only): text angle in degrees or `midangle` (default) for dynamic angles based on the mid-angle of the pie sector.
- **innerRadius:** For pie charts: defines the inner radius to create a *doughnut chart*.
- **formatjson:** format JSON object for better legibility

Examples

Line Chart:

```
{{Graph:Chart|width=400|height=100|type=line|x=1,2,3,4,5,6,7,8|y=10, 12, 6, 14, 2, 10, 7, 9}}
```

Note: The y-axis starts from the smallest y value, though this can be overridden with the `yAxisMin` parameter.

Area chart:

```
{{Graph:Chart|width=400|height=100|type=area|x=1,2,3,4,5,6,7,8|y=10, 12, 6, 14, 2, 10, 7, 9}}
```

Note: The y-axis starts from zero

Bar chart:



```
{{Graph:Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|type=rect|x=1,2,3,4,5,6,7,8|y=10, 12, 6, 14, 2, 10, 7, 9}}
```

Line chart with more than one data series, using colors:

```
{{Graph:Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|legend=Legend|type=line|x=1,2,3,4,5,6,7,8|y1=10, 12, 6, 14, 2, 10, 7, 9 |y2=2,4,6,8,13,11,9,2|colors=#0000aa,#ff8000}}
```

Area chart with more than one data series showing blended overlap:

```
{{Graph:Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|legend=Legend|type=area|x=1,2,3,4,5,6,7,8|y1=10, 12, 6, 14, 2, 10, 7, 9 |y2=2,4,6,8,13,11,9,2|colors=#800000aa,#80ff8000}}
```

Bar chart with multiple data series:

```
{{Graph:Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|legend=Legend|type=rect|x=1,2,3,4,5,6,7,8|y1=10, 12, 6, 14, 2, 10, 7, 9 |y2=2,4,6,8,13,11,9,2|colors=#800000aa,#80ff8000}}
```

Area chart with smoothed data values:



```
{{Graph:  
Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|legend=Legend|type=stackedarea:  
2,3,4,5,6,7,8|y1=10, 12, 6, 14, 2, 10, 7, 9 |y2=2,4,6,8,13,11,9,2  
|interpolate=monotone|colors=seagreen, orchid}}
```

Bar chart with stacked data series:

```
{{Graph:  
Chart|width=400|height=100|xAxisTitle=X|yAxisTitle=Y|legend=Legend|type=stackedrect  
2,3,4,5,6,7,8|y1=10, 12, 6, 14, 2, 10, 7, 9 |y2=2,4,6,8,13,11,9,2|y1Title=Data  
A|y2Title=Data B|colors=seagreen, orchid}}
```

Note that the order of the stacking is governed by the alphabetic value of the y-Titles used for the legend. You can always prepend a number 1, 2, 3 to establish whatever stacking order you want.



Vorlage:Graph:Chart/styles.css

```
/* {{pp-template}} */  
/* Prevent overflow behavior in mobile skins. */  
@media all and ( max-width: 720px ) {  
    body.skin-minerva .mw-graph {  
        min-width: 100% !important;  
        max-width: 100%;  
        overflow-x: auto;  
        overflow-y: visible;  
    }  
}  
  
.mw-graph-img {  
    width: inherit;  
    height: inherit;  
}
```