Inhaltsverzeichnis

Modul:Graph/Doku

Dies ist die Dokumentationsseite für Modul:Graph

- **basemap:** sets the base map. The map definitions must follow the TopoJSON format and if saved in Wikipedia are available for this module. Maps in the default directory Special: Prefixindex/Template:Graph:Map/Inner/ like Worldmap2c-json should only be referenced by their name while omitting the Module:Graph/ prefix to allow better portability. The parameter also accepts URLs, e.g. maps from other Wikipedia versions (the link should follow the scheme of `//en.wikipedia.org/w/index.php?title=`*`mapname`*`&action=raw`, i.e. protocol-relative without leading http/s and a trailing action=raw to fetch the raw content only). URLs to maps on external sites should be avoided for the sake of link stability, performance, security, and she be assumed to be blocked by the software or browser anyway.
- **scale:** the scaling factor of the map (default: 100)
- **projection:** the map projection to use. Supported values are listed at https://github.com/d3/d3-geo-projection. The default value is `equirectangular` for an equirectangular projection.
- **center:** map center (corresponds in the map data to both comma-separated values of the `scale` field)
- **feature:** which geographic objects should be displayed (corresponds in the map data to the name of the field under the `objects` field). The default is value `countries`.
- **ids of geographic entities**: The actual parameter names depend on the base map and the selected feature. For example, for the above mentioned world map the ids are ISO country codes. The values can be either colors or numbers in case the geographic entities should be associated with numeric data: `DE=lightblue` marks Germany in light blue color, and `DE=80.6` assigns Germany the value 80.6 (population in millions). In the latter case, the actual color depends on the following parameters.
- **colorScale:** the color palette to use for the color scale. The palette must be provided as a comma-separated list of color values. The color values must be given either as `#rgb`/`#rrggbb` or by a CSS color name. Instead of a list, the built-in color palettes `category10` and `category20` can also be used.
- **scaleType:** supported values are `linear` for a linear mapping between the data values and the color scale, `log` for a log mapping, `pow` for a power mapping (the exponent can be provided as `pow 0.5`), `sqrt` for a square-root mapping, and `quantize` for a quantized scale, i.e. the data is grouped in as many classes as the color palette has colors.
- **domainMin:** lower boundary of the data values, i.e. smaller data values are mapped to the lower boundary
- **domainMax:** upper boundary of the data values, i.e. larger data values are mapped to the upper boundary
- **legend:** show color legend (does not work with `quantize`)
- **defaultValue:** default value for unused geographic entities. In case the id values are colors the default value is `silver`, in case of numbers it is 0.
- **formatjson:** format JSON object for better legibility

chart

Creates a JSON object for `<graph>` to display charts. In the article namespace the template Template:Graph:Chart 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/nyurik/vega/wiki/Marks#line.
- **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 if n <= 10 is Category10: **Lua-Fehler: bad argument #1 to "get" (not a valid title)** else is Category20: **Lua-Fehler: bad argument #1 to "get" (not a valid title)**). See Template:ChartColors for details.
- **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 (not yet supported for bar charts). These parameters can be used to invert the scale of a numeric axis by setting the lowest value to the Max and highest value to the Min.
- **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. For example, the format `%` can be used to output percentages. For date/time specification of supported values is https://github.com/d3/d3-3.x-api-reference/blob/master/Time-Formatting.md , e.g. `xAxisFormat=%d-%m-%Y` for result 13-01-1977.
- **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 (use `string` to prevent axis values from being repeated when there are only a few values). Remarks: `Date` type doesn't work for bar graphs. For `date` data input please use ISO date format (e.g. YYYY-MM-DD) acc. to date and time formats used in HTML. Other date formats may work but not in all browsers. Date is unfortunately displayed only in en-US format for all Wikipedia languages. Workaround is to use **xAxisFormat** and **yAxisFormat** with numerical dates format.
- **xScaleType** and **yScaleType**: scale types of the x and y axes, e.g. `linear` for linear scale (default), `log` for logarithmic scale and `sqrt` for square root scale.

  > A logarithmic chart allows only positive values to be plotted. A square root scale chart cannot show negative values.

- **x**: the x-values as a comma-separated list, for dates and time see remark in **xType** and **yType**

- **y** or **y1**, **y2**, ...: the y-values for one or several data series, respectively. For pie charts `y2` denotes the radius of the corresponding sectors. For dates and time see remark in **xType** and **yType**
- **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. Setting to 0 with `type=line` creates a scatter plot.
- **linewidths**: different line widths may be defined for each series of data with csv, if set to 0 with "showSymbols" results with points graph, eg.: `linewidths=1, 0, 5, 0.2`
- **showSymbols**: show symbol on data point for line graphs, if number is provided it's size of symbol, default 2.5. may be defined for each series of data with csv, eg.: `showSymbols=1, 2, 3, 4`
- **symbolsShape**: custom shape for symbol: circle, x, square, cross, diamond, triangle_up, triangle_down, triangle_right, triangle_left. May be defined for each series of data with csv, eg.: `symbolsShape= circle, cross, square`
- **symbolsNoFill**: if true symbol will be without fill (only stroke),
- **symbolsStroke**: if "x" symbol is used or option "symbolsNoFill" symbol stroke width, default 2.5
- **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` (e.g. Vorlage:Para).
  - **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*.
- **xGrid** and **yGrid**: display grid lines on the x and y axes.
- **Annotations**
  - **vAnnotationsLine** and **hAnnotationsLine**: display vertical or horizontal annotation lines on specific values e.g. `hAnnotationsLine=4, 5, 6`
  - **vAnnotationsLabel** and **hAnnotationsLabel**: display vertical or horizontal annotation labels for lines e.g. `hAnnotationLabel = label1, label2, label3`
- **formatjson**: format JSON object for better legibility

Template wrappers

The functions `mapWrapper` and `chartWrapper` are wrappers to pass all parameters of the calling template to the respective `map` and `chart` functions.

Note: In the editor preview the graph extension creates a canvas element with vector graphics. However, when saving the page a PNG raster graphics is generated instead. {{#invoke:Graph|*function_wrapper_name*}}