

Warning: This API is deprecated. Please use the actively maintained [Google Charts \(/chart\)](#) API instead. See our [deprecation policy \(/chart/terms\)](#) for details.

Map Charts

This page describes how to create a colored map using the Charts API. This chart is a new, improved version that supports a larger map and more features than the [older map chart](https://developers.google.com/chart/image/docs/gallery/map_charts?hl=es) (`cht=t`).

Table of Contents

Overview

You can create a map with various countries or states highlighted in custom colors. You can let the map zoom to the proper level to display your selected countries properly, or you can zoom in or out to a custom level using latitude and longitude values.

Supported Parameters

Map charts support the following parameters:

Parameter	Required or Optional	Description
<code>cht=map<opt_zoom_and_center></code>	Required	Specifies a map chart. <code>opt_zoom_and_center</code> - An optional string that enables you to zoom or center the chart on a specific region. See details (#zoomcenter) .
<code>chs</code>	Required	Map size. Maximum dimensions for a map are 600 pixels wide, 600 pixels high, or 300,000 pixels total.
<code>chd=<first_country_value>, ... , <last_country_value></code>	Optional	If used, the data value will indicate the color of the corresponding country, along the color gradient specified by <code>chco</code> .
<code>chld=<country_or_region_codes> . . . </code>	Required	A pipe-separated list of countries or states that will be highlighted in colors specified by <code>chco</code> . The map will automatically zoom to include all the countries specified, unless you do custom zooming or centering (#zoomcenter) . Codes must be uppercase. These codes are ISO 3166-1-alpha-2 codes (http://en.wikipedia.org/wiki/ISO_3166-2): two uppercase characters, optionally followed by a dash and a second identifier for regions or cities. Examples: GB (Great Britain), GB-LND (London), US-NJ (state of New Jersey in the United States)
<code>chco</code> • If <code>chd</code> is not used: <code>chco=<unselected_land_color> <first_region_color> ... <last_region_color></code> • If <code>chd</code> is used: <code>chco=<unselected_land_color>, <color_range_start>, ... , <color_range_end></code>	Optional	Chart colors. The format of this string depends on whether you use <code>chd</code> . In both syntaxes, <code><unselected_land_color></code> is the color of any regions not specified in <code>chld</code> . Note that the dividers are different for the different syntaxes (commas in one, pipes in the other). If not used, regions will be outlined but not colored. • chd used - The second and later colors describe a color range. You must have one <code>chd</code> value for each country in <code>chld</code> , and the corresponding country will be colored along a range from <code><first_region_color></code> (minimum data range value) to <code><last_region_color></code> (maximum data range value). See examples in the table below. • chd not used - The second and later colors are applied to the corresponding regions in <code>chld</code> , in the order listed. If there are more countries than colors, the last color will be applied to all excess countries in the list.
<code>chtt, chts</code>	Optional	Chart title and style (<code>#gcharts_chart_title</code>)
<code>chm</code>	Optional	Chart markers (<code>#gcharts_shape_markers</code>)

chma	Optional Chart margins (#gcharts_chart_margins)
chf=bg...	Optional Solid fills (#gcharts_solid_fills) (background only)
chdl	Optional Chart legends (#gcharts_legend)

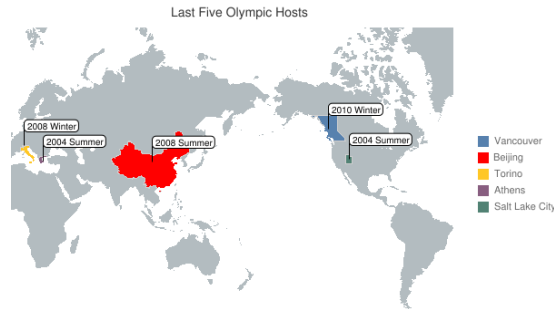
Examples

Description

Example

In this example:

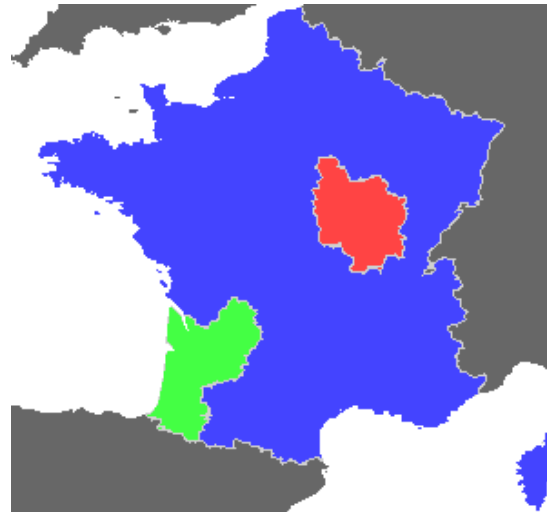
- `cht=map:fixed=-60,0,80,-35` - The chart zoom area is -60 latitude to 80 latitude, 0 longitude to -35 longitude
- `chld=CA-BC|CN|IT|GR|US-UT` - Regions highlighted are British Columbia, China, Italy, Greece, and Utah.
- `chm` - Chart markers for the flags.
- `chma=0,110,0,0` - The right side border increased to 110 pixels, to fit the legend properly.
- `chtt=Last+Five+Olympic+Hosts` - Chart title, with + used to indicate spaces.



```
cht=map:fixed=-60,0,80,-35
chs=600x350
chld=CA-BC|CN|IT|GR|US-UT
chdl=Vancouver|Beijing|Torino|Athens|Salt+Lake+City
chco=B3BCC0|5781AE|FF0000|FFC726|885E80|518274
chtt=Last+Five+Olympic+Hosts
chm=f2010+Winter,000000,0,0,10
f2008+Summer,000000,0,1,10
f2008+Winter,000000,0,2,10,1,-5:10
f2004+Summer,000000,0,3,10
f2004+Summer,000000,0,4,10
chma=0,110,0,0
```

A map showing France, with two regions highlighted:

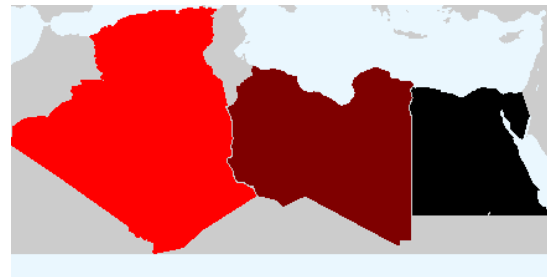
- `chld=FR-D|FR-B|FR` - Three regions are specified: Bourgogne (in the east), Aquitaine (in the southwest), and France.
- `chco=676767|FF4444|44FF44|4444FF` - No `chd` parameter is specified in this map, so each color value after the first is used to specify the corresponding country. Red for Bourgogne, green for Aquitaine, blue for France. You can see that later colors do not mask earlier colors.



```
chld=FR-D|FR-B|FR
chco=676767|FF4444|44FF44|4444FF
```

This map demonstrates using color gradients.

- `chd=t:0,50,100` - The `chd` parameter indicates a color gradient. Here, text encoding is specified, which has a range from 0–100.
- `chco=CCCCCC,FF0000,000000` - Background color is light gray (CCCCCC), and the color range is from red (FF0000) to black (000000). The middle country shown is an intermediate color on that gradient.



```
chd=t:0,50,100
chco=CCCCCC,FF0000,000000
```

Note that the map can be cropped inside the specified chart size. This is because we will not distort a map to fit inside a specified chart size, but we limit the map to the regions that you specify in `chld`. Therefore, to avoid cropped regions inside the chart boundaries, you might have to adjust the map size manually.



cht=map
chs=300x300
chld=GB

England is naturally long and thin, so specifying a square map includes a cropped area on the right side:

cht=map
chs=180x300
chld=GB

Adjust the chart width to crop off the excess. Note how there's still some cropping on the bottom that should be fixed by reducing the map height.

cht=map
chs=180x300
chld=GB

France is much more regularly shaped, so it fits more naturally in a square map.

[Back to top \(#top\)](#)

Zooming and Centering

By default, the chart will be zoomed and centered to center your selected regions, and make them fit into the rendered area as closely as possible, within the defined map size. However, you can customize the centering and zooming level of the chart in two different ways:

By pixel

To adjust the borders by pixel size, use this format:

cht=map:auto=<left_border>,<right_border>,<top_border>,<bottom_border>

Where each border size is the additional border to show, in pixels, beyond the automatically calculated zoom level.

You can only zoom out (positive values); you cannot zoom in (negative values) closer than the default zoom level using pixel zooming.

Examples:

These examples all show London (Lat

No Zooming (default)



cht=map

Zoomed out 30 pixels per side



cht=map:auto=30,30,30,30

Zoomed out and down to the right



cht=map:auto=50,0,80,0

By Latitude and Longitude

You can specify the exact top, bottom, left, and right borders of the rendered map by latitude and longitude using the syntax shown here:

```
cht=map:fixed=<bottom_border_lat>,<left_border_long>,<top_border_lat>,<right_border_long>
```

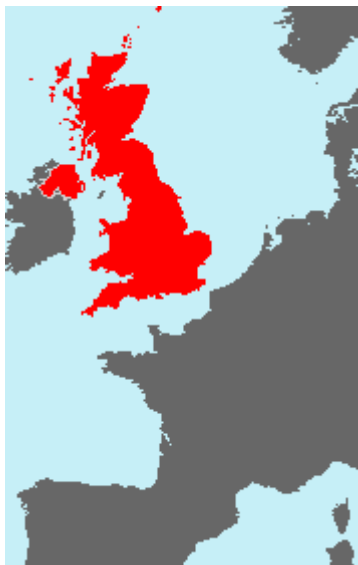
Where `<bottom_border_lat>` and `<top_border_lat>` specify the latitude of the bottom and top borders, and `<left_border_long>` and `<right_border_long>` specify the longitude of the left and right borders.

Note: Because of Mercator projection distortion (basically, projecting a curved map onto a flat image), focusing on a tall, narrow strip can cause some unexpected size differences in your map.

Examples:

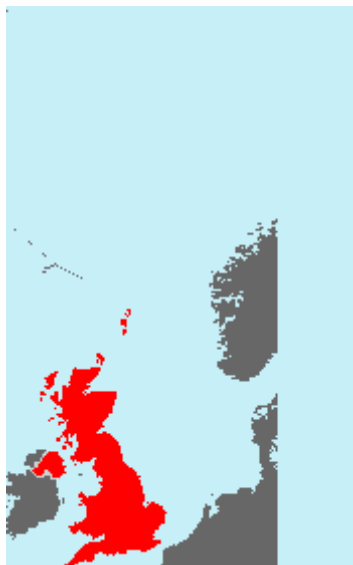
The following examples show London (highlighted in blue) which is 0.1° longitude, 51.5° latitude. Note how there is map cropping because the map is trying to display only Great Britain (as specified) and fill the designated chart without distortion.

Centered



```
cht=map:fixed=40,-10,60,10
```

Latitudes shifted +10*



```
cht=map:fixed=50,-10,70,10
```

Zoomed by reducing all values



```
cht=map:fixed=48,-3,54,3
```

Note how the size of the middle image is a bit smaller than the first image, even though we've moved the same number of degrees up. This is because we're projecting a 3-D sphere onto a 2-D image. You would have to increase the width of the second chart to make it the same size as the first chart.

[Back to top \(#top\)](#)

Standard Features

The rest of the features on this page are standard chart features.

Chart Title `chtt`, `chts` [*All charts*]

You can specify the title text, color, and font size for your chart.

Syntax

```
chtt=<chart_title>  
chts=<color>,<font_size>,<opt_alignment>
```

`chtt` - Specifies the chart title.

`<chart_title>`

Title to show for the chart. You cannot specify where this appears, but you can optionally specify the font size and color. Use a `+` sign to indicate spaces, and a pipe character (`|`) to indicate line breaks.

chts [Optional] - Colors and font size for the `cht` parameter.

<color>

The title color, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). *Default color is black.*

<font_size>

Font size of the title, in points.

<opt_alignment>

[Optional] Alignment of the title. Choose one of the following case-sensitive string values: "l" (left), "c" (centered) "r" (right). *Default is "c".*

Examples

Description

A chart with a title, using default color and font size.

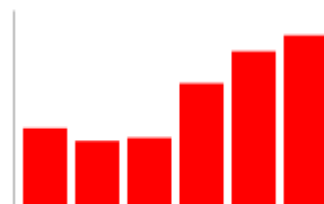
Specify a space with a plus sign (+).

Use a pipe character (|) to force a line break.

`chts` is not specified here.

Example

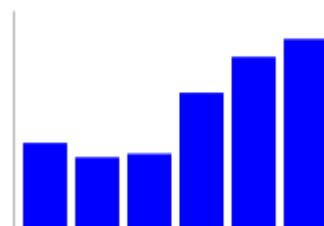
Site visitors by month
January to July



```
cht=Site+visitors+by+month|  
January+to+July
```

A chart with a blue, right-aligned, 20-point title.

Site visitors



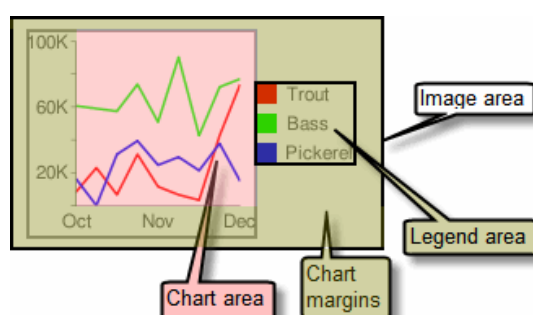
```
cht=Site+visitors  
chts=FF0000,20,r
```

[Back to top \(#top\)](#)

Chart Margins `chma` [All charts]

You can specify the size of the chart's margins, in pixels. Margins are calculated inward from the specified chart size (`chs`); increasing the margin size does not increase the total chart size, but rather shrinks the chart area, if necessary.

The margins are by default whatever is left over after the chart size is calculated. This default value varies by chart type. The margins that you specify are a *minimum* value; if the chart area leaves room for margins, the margin size will be whatever is left over; you cannot squeeze the margins smaller than what is required for any legends and labels. Here's a diagram showing the basic parts of a chart:



The *chart margins* include the *axis labels* and the *legend area*. The legend area resizes automatically to fit the text exactly, unless you specify a larger width using `chma`, in which case it will expand the margin size wider, squeezing the chart area smaller. You cannot crop a legend by specifying a size that is too small, but you can make it take up more space than it needs.

Tip: In a bar chart, if the bars have a fixed size (the default), the chart area width cannot be reduced. You must specify a smaller or resizable bar size using [chbh](#) (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es#chbh).

Syntax

chma=

<left_margin>, <right_margin>, <top_margin>, <bottom_margin> | <opt_legend_width>, <opt_legend_height>

<left_margin>, <right_margin>, <top_margin>, <bottom_margin>

Minimum margin size around the chart area, in pixels. Increase this value to include some padding to prevent axis labels from bumping against the borders of the chart.

<opt_legend_width>, <opt_legend_height>

[Optional] Width of the margin around the legend, in pixels. Use this to avoid having the legend bump up against the chart area or the edges of the image.

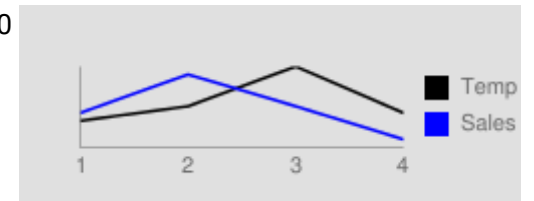
Examples

Description

In this example, the chart has a minimum margin of 30 pixels on each side. Because the chart legend is more than 30 pixels wide, the margin on the right side is set to the width of the chart legend, and is different from the other margins.

Axis labels are outside the plot area, and are therefore drawn within the margin space.

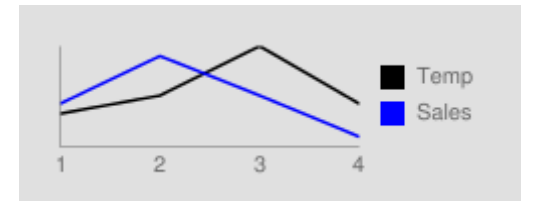
Example



chma=30, 30, 30, 30

To add a margin around the legend, set a value for the <opt_legend_width> and <opt_legend_height> parameters.

In this example, the legend is approximately 60 pixels wide. If you set the <opt_legend_width> to 80 pixels, the margin extends to 20 pixels outside the legend.



chma=20, 20, 20, 30 | 80, 20

[Back to top \(#top\)](#)

Background Fills chf [All charts]

You can specify fill colors and styles for the chart data area and/or the whole chart background. Fill types include solid fills, striped fills, and gradients. You can specify different fills for different areas (for example, the whole chart area, or just the data area). The chart area fill overwrites the background fill. All fills are specified using the chf parameter, and you can mix different fill types (solids, stripes, gradients) in the same chart by separating values with pipe character (|). Chart area fills overwrite chart background fills.

Solid Fills chf [All Charts]

You can specify a solid fill for the background and/or chart area, or assign a transparency value to the whole chart. You can specify multiple fills using the pipe character (|). (Maps: background only).

Syntax

chf=<fill_type>, s, <color> | ...

<fill_type>

The part of the chart being filled. Specify one of the following values:

- **bg** - Background fill
- **c** - Chart area fill. *Not supported for map charts.*
- **a** - Make the whole chart (including backgrounds) transparent. The first six digits of <color> are ignored, and only the last two (the transparency value) are applied to the whole chart and all fills.
- **b<index>** - Bar solid fills (bar charts only). Replace <index> with the series index of the bars to fill with a solid color. The effect is similar to specifying chco in a bar chart. See [Bar Chart Series Colors](https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) for an example.

s

Indicates a solid or transparency fill.

<color>

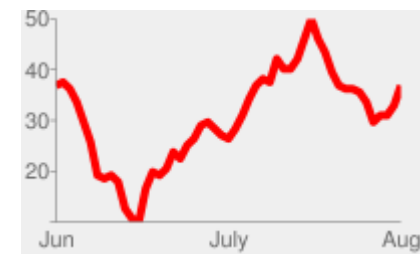
The fill color, in [RRGGBB hexadecimal format](https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb) (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). For transparencies, the first six digits are ignored, but must be included anyway.

Examples

Description

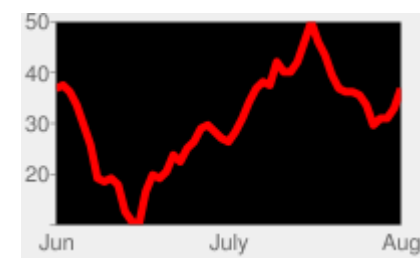
This example fills the chart background with pale gray (EFEFEF).

Example



```
chf=bg,s,EFEFEF
```

This example fills the chart background with pale gray (EFEFEF) and fills the chart area in black (000000).



```
chf=c,s,000000|  
bg,s,EFEFEF
```

This example applies a 50% transparency to the whole chart (80 in hexadecimal is 128, or about 50% transparency). Notice the table cell background showing through the chart.



```
chf=a,s,00000080
```

[Back to top](#) (#top)

Chart Legend Text and Style chd1, chd1p, chd1s [All charts]

The legend is a side section of the chart that gives a small text description of each series. You can specify the text associated with each series in this legend, and specify where on the chart it should appear.

See also [chma](#) (#gcharts_chart_margins), to learn how to set the margins around your legend.

A note on string values: Only URL-safe characters are permitted in label strings. To be safe, you should URL-encode any strings containing characters not in the character set `0-9a-zA-Z`. You can find a URL encoder in the [Google Visualization Documentation](https://developers.google.com/chart/interactive/docs/querylanguage?hl=es#plainText) (<https://developers.google.com/chart/interactive/docs/querylanguage?hl=es#plainText>).

Syntax

```
chd1=<data_series_1_label>|...|<data_series_n_label>  
chd1p=<opt_position>|<opt_label_order>  
chd1s=<color>,<size>
```

chd1 - The text for each series, to display in the legend.

<data_series_label>

The text for the legend entries. Each label applies to the corresponding series in the `chd` array. Use a + mark for a space. If you do not specify this parameter, the chart will not get a legend. There is no way to specify a line break in a label. The legend will typically expand to hold your legend text, and the chart area will shrink to accommodate the legend.

chd1p - [Optional] The position of the legend, and order of the legend entries. You can specify <position> and/or <label_order>. If you specify both, separate them with a bar character. You can add an 's' to any value if you want empty legend entries in `chd1` to be skipped in the legend. Examples: `chd1p=bv`, `chd1p=r`, `chd1p=bv|r`, `chd1p=bvs|r`

<opt_position>

[Optional] Specifies the position of the legend on the chart. To specify additional padding between the legend and the chart area or the image border, use the `chma` (#gcharts_chart_margins) parameter. Choose one of the following values:

- `b` - Legend at the bottom of the chart, legend entries in a horizontal row.
- `bv` - Legend at the bottom of the chart, legend entries in a vertical column.
- `t` - Legend at the top of the chart, legend entries in a horizontal row.
- `tv` - Legend at the top of the chart, legend entries in a vertical column.
- `r` - [Default] Legend to the right of the chart, legend entries in a vertical column.
- `l` - Legend to the left of the chart, legend entries in a vertical column.

<opt_label_order>

[Optional] The order in which the labels are shown in the legend. Choose one of the following value:

- `l` - [Default for vertical legends] Display labels in the order given to `chd1`.
- `r` - Display labels in the reverse order as given to `chd1`. This is useful in stacked bar charts to show the legend in the same order as the bars appear.
- `a` - [Default for horizontal legends] Automatic ordering: roughly means sorting by length, shortest first, as measured in 10 pixel blocks. When two elements are the same length (divided into 10 pixel blocks), the one listed first will appear first.
- `0, 1, 2, ...` - Custom label order. This is a list of zero-based label indexes from `chd1`, separated by commas.

chd1s - [Optional] Specifies the color and font size of the legend text.

<color>

The legend text color, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb).

<size>

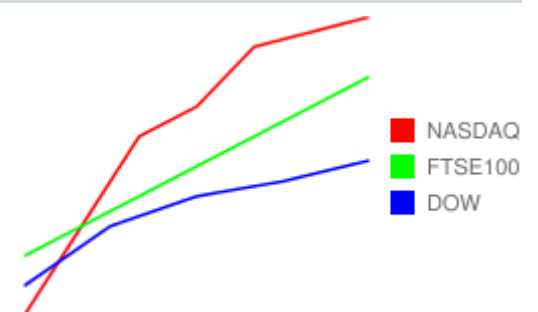
The point size of the legend text.

Examples

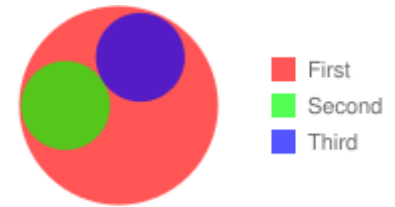
Description

Two examples of legends. Specify legend text in the same order as your data series.

Example



`chd1=NASDAQ|FTSE100|DOW`
`chco=FF0000,00FF00,0000FF`



```
chdl=First|Second|Third
chco=ff0000,00ff00,0000ff
```

The first chart demonstrates horizontal legend entries (`chdlp=t`, default layout is horizontal), and the second demonstrates bottom vertical legend entries (`chdlp=bv`).

```
■ First ■ Third ■ Second
```



```
chdl=First|Second|Third
chco=ff0000,00ff00,0000ff
chdlp=t
```



```
■ First
■ Second
■ Third
```

```
chdl=First|Second|Third
chco=ff0000,00ff00,0000ff
chdlp=bv
```

This example demonstrates changing the font size.

```
■ First ■ Third ■ Second
```



```
chdls=0000CC,14
```

[Back to top \(#top\)](#)

Shape Markers `chm` [*Bar, Line, Radar, Scatter*]

You can specify graphical markers for all or individual data points on a chart. If two or more markers occupy the same point, the markers are drawn in the order in which they appear in the `chm` parameter. You can also create text markers on data points, which is covered in [Data Point Markers](https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_data_point_labels) (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_data_point_labels).

You can combine shape markers with any other `chm` parameters using a pipe character (`|`) to separate the `chm` parameters.

Syntax

Specify one set of the following parameters for each series that should be marked. To mark multiple series, create additional parameter sets, delimited by a pipe character. You do not need to mark up all series. If you do not assign markers to a data series, it will not get any markers.

Shape markers behave slightly differently in [scatter charts](#)

(https://developers.google.com/chart/image/docs/gallery/scatter_charts?hl=es#chart_types). See that documentation for more information.

```
chm=
  [ @ ] <marker_type>, <color>, <series_index>, <opt_which_points>, <size>, <opt_z_order>, <opt_offset>
  | ... |
  [ @ ] <marker_type>, <color>, <series_index>, <opt_which_points>, <size>, <opt_z_order>, <opt_offset>
```

@

[Optional] If you precede the marker type with the optional @ character, then `<opt_which_points>` should use the `x:y` format.

<marker_type>

The type of marker to use. Specify one of the following types:

- a - Arrow
- c - Cross
- C - Rectangle. If a rectangle marker, you must have at least two data series, where series 0 specifies the bottom edge and series 1 specifies the top edge. `<size>` specifies the width of the rectangle, in pixels.
- d - Diamond
- E - Error-bar marker (\perp) This marker requires two data series to create, one value for the bottom, and the corresponding point in the second series for the top. It also exposes an extended `<size>` syntax: `line_thickness[:top_and_bottom_width]` where `top_and_bottom_width` is optional. See the examples below.
- h - Horizontal line across the chart at a specified height. (**The only valid format for `<opt_which_points>` parameter is `n.d.`**)
- H - Horizontal line through the specified data marker. This supports an extended `<size>` syntax that lets you specify an exact line length: `line_thickness[:length]` where `:length` is optional, and defaults to the full chart area width.
- o - Circle
- s - Square
- v - Vertical line from the x-axis to the data point
- V - Vertical line of adjustable length. This supports an extended `<size>` value syntax that lets you specify an exact line length: `line_thickness[:length]` where `:length` is optional, and defaults to the full chart area height. The marker is centered on the data point.
- x - An X

<color>

The color of the markers for this series, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb).

<series_index>

The zero-based index of the data series on which to draw the markers. Ignored for h markers and markers that specify location by x/y position (start with the @ character). You can use hidden data series as a source for markers; see Compound Charts (https://developers.google.com/chart/image/docs/gallery/compound_charts?hl=es) for more information. Grouped vertical bar charts support a special extended syntax (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es#markerplacementtarget) to align markers with specific bars.

<opt_which_points>

[Optional] Which point(s) to draw markers on. *Default is all markers.* Use one of the following values:

- *n.d* - Where to draw the marker. The meaning depends on the marker type:
 - *All types except h* - Which data point to draw the marker on, where *n.d* is the zero-based index in the series. If you specify a non-integer value, then the fraction indicates a calculated intermediate point. For example, 3.5 means halfway between point 3 and point 4.
 - *h* - A number from 0.0 to 1.0, where 0.0 is the bottom of the chart, and 1.0 is the top of the chart.
- -1 - Draw a marker on all data points. You can also leave this parameter empty to draw on all data points.
- -*n* - Draw a marker on every *n*-th data point. Floating point value; if *n* is less than 1 the chart will calculate additional intermediary points for you. For example, -0.5 will put twice as many markers as data points.
- *start:end:n* - Draw a marker on every *n*-th data point in a range, from *start* to *end* index values, inclusive. All parameters are optional (may be absent), so *3::1* would be from the fourth element to the last, step 1, and omitting this parameter entirely would default to *first:last:1*. All values can be floating point numbers. *start* and *end* can be negative, to count backward from the last value. If both *start* and *end* are negative, be sure that they are listed in increasing value (for example, -6:-1:1). If the *n* step value is less than 1, it will calculate additional data points by interpolating the data values given. *Default values are first:last:1*
- *x:y* - Draw a marker at a specific x/y point on the chart. This point does not have to be on a line. Add the @ character before the marker type to use this option. Specify the coordinates as floating point values, where 0:0 is the bottom left corner of the chart and 1:1 is the top right corner of the chart. For example, to add a red, 15-pixel diamond to the center of a chart, use @d,FF0000,0,0.5:0.5,15.

<size>

The size of the marker, in pixels. Most take a single number value for this parameter; the V, H, and S markers support the syntax `<size>[:width]` where the optional second part specifies the line or marker length.

<opt_z_order>

[Optional] The layer on which to draw the marker, compared to other markers and all other chart elements. This is a floating point number from -1.0 to 1.0, inclusive, where -1.0 is the bottom and 1.0 is the top. Chart elements (lines and bars) are just lower than zero. If two markers have the same value, they are drawn in the order given by the URL. *Default value is 0.0* (just above the chart elements).

<opt_offset>

[Optional] Let you specify horizontal and vertical offsets from the specified location. Here is the syntax, which uses a : delimiter: `reserved:<horizontal_offset>:<vertical_offset>`. If specified, you can include an empty ,, value in the chm parameter string for <opt_z_order>. Examples: `o,FF9900,0,4,12,, :10` `o,FF9900,0,4,12.0,, :-10:20` `o,FF9900,0,4,12,1, ::20`

- `reserved` - Leave blank.
- `<horizontal_offset>` - A positive or negative number specifying the horizontal offset, in pixels. Optional; leave blank if not used.
- `<vertical_offset>` - A positive or negative number specifying the vertical offset, in pixels. Optional; leave blank if not used.

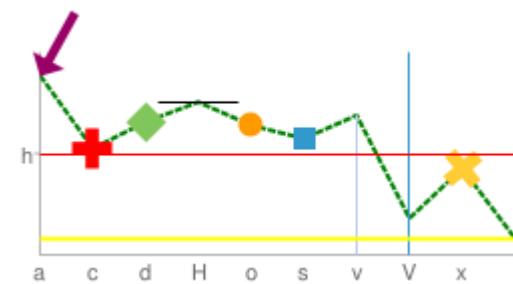
Examples

Description

Here's an example of several of the shape and line markers.

- `a, 990066, 0, 0.0, 9.0` - Purple arrow, first series, first point, size 9.
- `c, FF0000, 0, 1.0, 20.0` - Red cross, first series, second point, size 9.
- `d, 80C65A, 0, 2.0, 20.0` - Green diamond, first series, third point, size 9.
- `H, 000000, 0, 3, 1:40` - Black horizontal line, first series, data point 3, one pixel wide, forty pixels long.
- `o, FF9900, 0, 4.0, 12.0` - Orange circle, first series, fifth point, size 12.
- `s, 3399CC, 0, 5.0, 11.0` - Blue square, first series, sixth point, size 11.
- `v, BBCCED, 0, 6.0, 1.0` - Vertical line up to point, first series, seventh point, one pixel wide.
- `V, 3399CC, 0, 7.0, 1.0` - Vertical line bottom to top of chart, first series, eighth point, one pixel wide.
- `x, FFCC33, 0, 8.0, 20.0` - Yellow 'X', first series, ninth point, size 20.
- `H, FFFF00, 0, 9, 2` - Horizontal yellow line the width of the chart at data point 9.
- `h, FF0000, 0, 0.5, 1` - Red horizontal line at designated height, first series, halfway up the chart, one pixel wide.

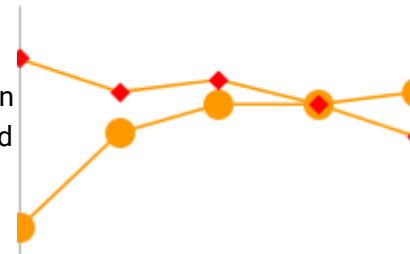
Example



```
chm=
a, 990066, 0, 0.0, 9.0 |
c, FF0000, 0, 1.0, 20 |
d, 80C65A, 0, 2.0, 20.0 |
H, 000000, 0, 3, 1:40 |
o, FF9900, 0, 4.0, 12.0 |
s, 3399CC, 0, 5.0, 11.0 |
v, BBCCED, 0, 6, 1.0 |
V, 3399CC, 0, 7, 1.0 |
x, FFCC33, 0, 8, 20 |
H, FFFF00, 0, 9, 2 |
h, FF0000, 0, 0.5, 1
```

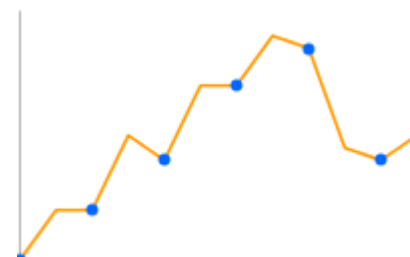
Here's an example using diamonds for one data series, and circles for the other data series.

If two or more markers occupy the same point, the markers are drawn in the order in which they appear in the `chm` parameter. Here, the circle is the first marker specified with `chm`, so it is drawn first. The diamond is specified and drawn second, which results in it being drawn on top of the circle.



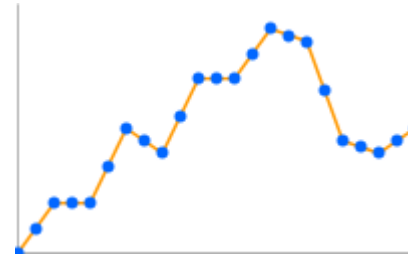
```
chm=
o, FF9900, 0, -1, 15.0 |
d, FF0000, 1, -1, 10.0
```

Here's a line chart with a marker on every second data point (-2 means every other point).



```
chd=t:
0, 20, 20, 50, 40, 70, 70, 90, 85, 45, 40, 50
chm=
o, 0066FF, 0, -2, 6
```

Here's a line chart with twice as many markers as data points (-0.5 means every half point).



```
chd=t:
  0,20,20,50,40,70,70,90,85,45,40,50
chm=
  o,0066FF,0,-.5,6
```

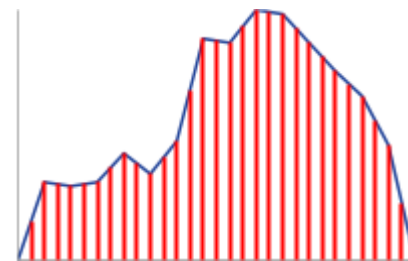
This example shows how to use h and v markers to create grid lines with custom colors and thickness. The z-order value (the last value) is set to -1 so that the grid lines are drawn beneath the data line.



```
chm=
  h,76A4FB,0,0:1:.2,2,-1|
  V,76A4FB,0,::2,0.5,-1
```

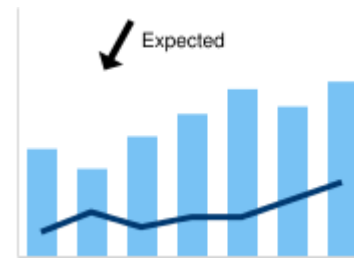
This chart adds vertical fill lines to a line chart:

- v - Vertical lines to the chart
- FF0000 - Red lines
- 0 - Series index
- : :.5 - Range specifier: from start to end, every 0.5 points.
- 2 - Thickness 2 pixels.



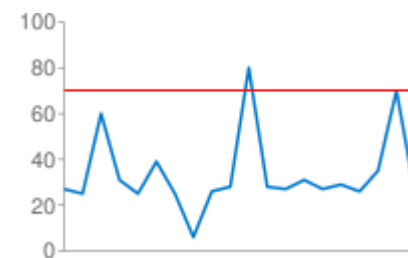
```
chm=
  v,FF0000,0,::.5,2
```

This example adds an arrow and text marker to the chart using exact coordinates. The first D marker is the trace line under the bars. The second marker is the arrow, and the third marker is the arrow text.



```
chm=
  D,003971,1,0,3|
  @a,000000,0,.25:.75,7|
  @tExpected,000000,0,.35:.85,10
```

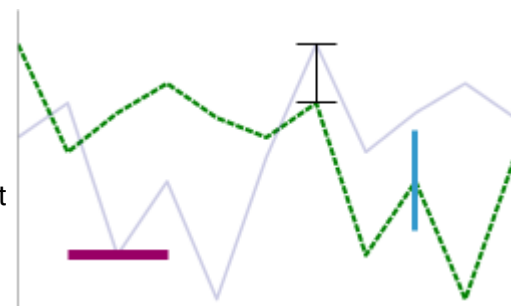
A horizontal line fixed to a specific data point (H) can be useful for showing relative values, or emphasizing the height of a data value on a chart.



```
chm=H,FF0000,0,18,1
```

This graph demonstrates the markers that can specify line thickness and length in the <size> parameter.

- E,000000,0,6,1:20 - Black error bar with 1 pixel wide lines, top and bottom bars 20 pixels long. The bottom is anchored to series 0 point 8, the top is anchored to series 1 point 8.
- H,990066,1,2,5:50 - Purple, horizontal line five pixels wide, fifty pixels long centered on data point 2.
- V,3399CC,0,8,3:50 - Blue, vertical line 3 pixels wide, fifty pixels long, centered on data point 8.



```
chm=
  E,000000,0,6,1:20|
  H,990066,1,2,5:50|
  V,3399CC,0,8,3:50
```

[Back to top \(#top\)](#)

Text and Data Value Markers chm [*Bar, Line, Radar, Scatter*]

You can label specific points on your chart with custom text, or with formatted versions of the data at that point.

You can combine any `chm` markers using a pipe character (`|`) to separate the `chm` parameter sets.

A note on string values: Only URL-safe characters are permitted in label strings. To be safe, you should URL-encode any strings containing characters not in the character set `0-9a-zA-Z`. You can find a URL encoder in the [Google Visualization Documentation](https://developers.google.com/chart/interactive/docs/querylanguage?hl=es#plainText) (<https://developers.google.com/chart/interactive/docs/querylanguage?hl=es#plainText>).

Syntax

Specify one set of the following parameters for each series that should be marked. To mark multiple series, create additional parameter sets, delimited by a pipe character. You do not need to mark up all series. If you do not assign markers to a data series, it will not get any markers.

```
chm=  
<marker_type>,<color>,<series_index>,<opt_which_points>,<size>,<opt_z_order>,<opt_placement>  
|...|  
<marker_type>,<color>,<series_index>,<opt_which_points>,<size>,<opt_z_order>,<opt_placement>
```

<marker_type>

The type of marker to use. You can choose from the following types:

- `f<text>` - A flag containing text. Specify the character 'f', followed by custom URL-encoded text.

★ To escape commas in text markers, precede the comma by a `\` mark.

Example: `fHello\,+World!`

- `t<text>` - A simple text marker. Specify the character 't' followed by custom URL-encoded text.

★ To escape commas in text markers, precede the comma by a `\` mark.

Example: `tHello\,+World!`

- `A<text>` - An annotation marker. This is similar to a flag marker, but markers will coordinate their position so that they do not overlap. The only valid format for `<opt_which_points>` is `n.d`, to signify the index of a point in the series.
- `N<formatting_string>` - The value of the data at this point, with optional formatting. If you do not use the `chds` (https://developers.google.com/chart/image/docs/data_formats?hl=es#data_scaling) parameter (custom scaling) it gives the exact encoded value; if you do use that parameter *with any format type* the value will be scaled to the range that you specify. See an example of `chds` with numeric markers below. With this marker type in a [stacked bar chart](https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es), if you specify `-1` for `<series_index>` you will get a marker that shows the sum of all values in this stacked bar. The formatting string syntax is as follows:

`<preceding_text>*<number_type><decimal_places>zs<x or y>*<following_text>`.

All of these elements are optional. Here is the meaning of each element:

- `<preceding_text>` - Text to precede each value.
- `*...*` - An optional block wrapped in literal asterisks, in which you can specify formatting details for numbers. The following values are supported, and are all optional:
 - `<number_type>` - The number format, for numeric values. Choose one of the following:
 - `f` - [Default] Floating point format. Consider specifying precision as well with the `<decimal_places>` value.
 - `p` - Percentage format. A % sign is appended automatically. **Note:** When using this format, data values from 0.0 – 1.0 map to 0 – 100% (for example, 0.43 will be shown as 43%).
 - `e` - Scientific notation format.
 - `c<CUR>` - Format the number in the currency specified, with the appropriate currency marker. Replace `<CUR>` with a three-letter currency code. Example: `cEUR` for Euros. You can find a list of codes on the [ISO web site](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=46121) (http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=46121), although not all symbols are supported.
 - `<decimal_places>` - An integer specifying how many decimal places to show. The value is rounded (not truncated) to this length. *Default is 2.*

- *z* - Display trailing zeros. *Default is no.*
- *s* - Display group separators. *Default is no.*
- *x* or *y* - Display the data from the *x*- or *y*-coordinate, as specified. The meaning of *x* data varies by chart type: experiment with your chart to determine what it means. *Default is 'y'.*
- *<following_text>* - Text to follow each value.

<color>

The color of the markers for this set, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb).

<series_index>

The zero-based index of the data series on which to draw the markers. If this is a stacked bar chart (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) and the marker type is *N* (data point values), you can specify *-1* to create a summed value marker for each stack of bars. See below for an example.

<opt_which_points>

[*Optional*] Which point(s) to draw markers on. *Default is all markers.* Use one of the following values:

- *n.d* - Which data point to draw the marker on, where *n.d* is the zero-based index in the series. If you specify a non-integer value, then the fraction indicates a calculated intermediate point. For example, 3.5 means halfway between point 3 and point 4.
- *-1* - Draw a marker on all data points. You can also leave this parameter empty to draw on all data points.
- *-n* - Draw a marker on every *n*-th data point.
- *start:end:n* - Draw a marker on every *n*-th data point in a range, from *start* to *end* index values, inclusive. All parameters are optional (may be absent), so *3::1* would be from the fourth element to the last, step 1, and omitting this parameter entirely would default to *first:last:1*. All values can be floating point numbers. *start* and *end* can be negative, to count backward from the last value. If both *start* and *end* are negative, be sure that they are listed in increasing value (for example, *-6:-1:1*). If the *n* step value is less than 1, it will calculate additional data points by interpolating the data values given. *Default values are first:last:1*
- *x:y* - [*Not supported for N-type markers*] Draw a marker at a specific *x/y* point on the chart. This point does not have to be on a line. Add the at character (@) before the marker type to use this option. Specify the coordinates as floating point values, where *0:0* is the bottom left corner of the chart, *0.5:0.5* is the center of the chart, and *1:1* is the top right corner of the chart. For example, to add a red, 15-pixel diamond to the center of a chart, use *@d, FF0000, 0, 0.5:0.5, 15*.

<size>

The size of the marker in pixels. If this is a scatter chart with a third data series (used to specify point sizes), this value will be scaled by the data range. So if the data range is 0–100 and *<size>* is 30, a data value of 100 would be 30 pixels wide, a data value of 50 would be 15 pixels wide, and so on.

<opt_z_order>

[*Optional*] The layer on which to draw the marker, compared to other markers and all other chart elements. This is a floating point number from -1.0 to 1.0, inclusive, where -1.0 is the bottom and 1.0 is the top. Chart elements (lines and bars) are just lower than zero. If two markers have the same value, they are drawn in the order given by the URL. Default value is 0.0 (just above the chart elements).

<opt_placement>

[*Optional*] Additional placement details describing where to put this marker, in relation to the data point. You can specify horizontal and/or vertical relative positioning, as well as offsets. Placement syntax is a string with : delimiters as shown here. All elements are optional: *<horizontal_and_vertical_justification>:<horizontal_offset>:<vertical_offset>*. If specified, you can include an empty ,, value in the *chm* parameter string for *<opt_z_order>*. Examples: *N, 000000, 0, 1, 10, ,, b* and *N, 000000, 0, 1, 10, ,, 1v* and *N, 000000, 0, 1, 10, ,, r::10*.

horizontal_and_vertical_justification

The anchor point of the marker. This behaves opposite to justification, so a left anchor actually puts the marker to the *right* of the data point. You can choose a horizontal and/or vertical justifier from the following list:

- **Horizontal placement:** 'l', 'h', or 'r' - Left, center, or right-anchored, horizontally. *Default is 'l'.*

- **Vertical placement:** 'b', 'v', 't' - Bottom, middle, or top-anchored, vertically. *Default is 'b'.*
- **Bar-relative placement [Bar charts only]:** 's', 'c', 'e' - Base, center, or top of a bar. For stacked charts, this is relative to the section of the bar for each series, not for the whole bar. If the series index given is -1 (stack total) it is in relation to the whole bar. This can be combined with vertical placement values: for example, 'be' or 'vs'. *Default value is 'e'.*

horizontal_offset

A horizontal offset for this marker, in pixels. *Default is 0.*

vertical_offset

A vertical offset for this marker, in pixels. *Non-bar chart default: 15; bar chart default: 2.*

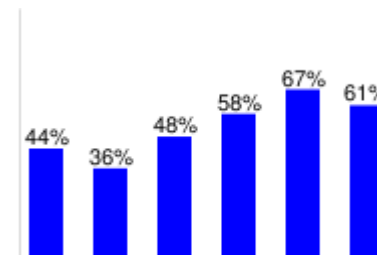
Description

Here's an example of value labels on a bar chart.

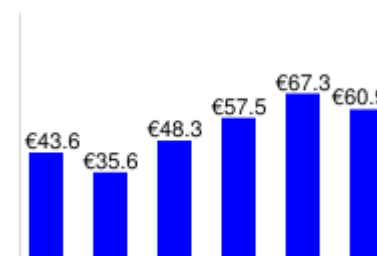
The first chart (**N*p0***) specifies a data value marker, with values shown as a percentage, rounded to zero decimal places, in black, on all values, in 11-point text. Note that the data values are all between 0.0 and 1.0, which, in percentage format, are moved up two decimal places.

The second chart (**N*cEUR1***) shows the same values formatted as Euro values, one decimal place, in black, on all values, in 11-point text.

Example



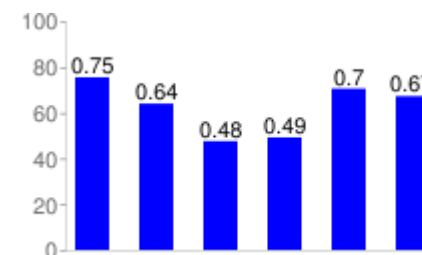
```
chm=
  N*p0*, 000000, 0, -1, 11
chd=t:
  0.4356, 0.3562, 0.4834, 0.575, 0.673, 0.6091
```



```
chm=
  N*cEUR1*, 000000, 0, -1, 11
```

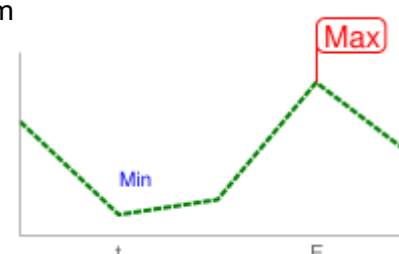
You can use the **chds** parameter to scale the numeric value displayed. You can use **chds** for all data formats, including simple and extended encoding. This will not affect the bar size or the axis labels, but only the data marker value.

This example shows a chart with simple-encoded values of 46, 39, 29, 30, 43, 41. Simple encoding range is 0–61. The **chds** parameter defines a custom marker scale of 0–1, so marker values are scaled to this range, but the bar heights are not affected (if this were [text format data](https://developers.google.com/chart/image/docs/data_formats?hl=es#data_scaling) (https://developers.google.com/chart/image/docs/data_formats?hl=es#data_scaling), the bars would be scaled as well).



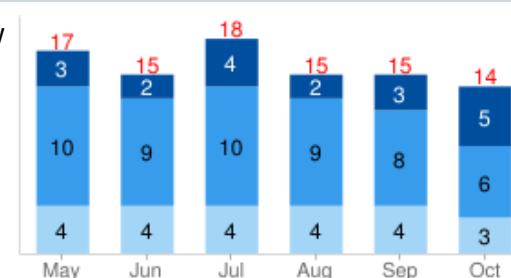
```
chd=s: underp
chm=N, 000000, 0, -1, 11
chds=0, 1
```

Here's an example of a chart with a text label at the minimum point and a flag label at the maximum point.



```
chm=
  tMin, 0000FF, 0, 1, 10 |
  fMax, FF0000, 0, 3, 15
```

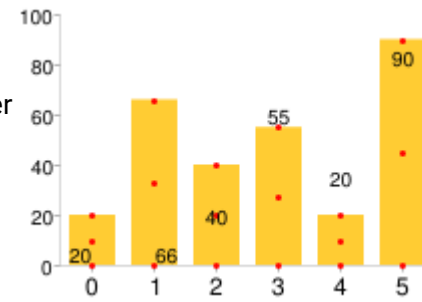
This example shows a stacked chart with values for individual series, plus the series total. To show the stacked series values, we must use the 'c' positioning option; if we did not, the top bar value would overlap the sum value at the top of each bar.



```
chm=
  N, FF0000, -1, , 12 |
  N, 000000, 0, , 12, , c |
```

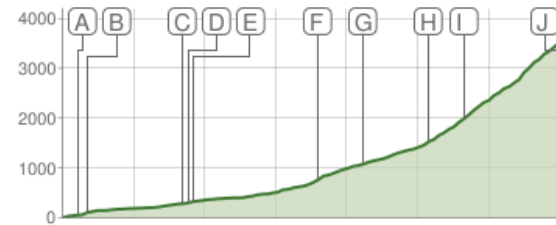
N,000000,1,,12,,c|
 N,ffffff,2,,12,,c

Some more demonstrations of how vertical and horizontal placement work. This example demonstrates various combinations of anchor values for bar charts (which use s, c, and e for vertical placement). Note how a right anchor moves a marker left, and a top anchor moves a marker down, and vice-versa. The red dots show the base, center, and top of each bar. The number is the data value, fixed using different anchor values for each bar.



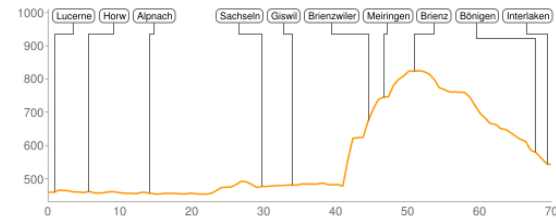
chm=
 N,000000,0,0,10,,rs
 N,000000,0,1,10,,ls
 N,000000,0,2,10,,c
 N,000000,0,3,10,,e
 N,000000,0,4,10,,e::15
 N,000000,0,5,10,,e::-12

Annotation markers automatically adjust the label position so that they don't overlap. The first chm value is for the line fill, the following values are all annotation markers.



chm=B,C5D4B5BB,0,0,0
 AA,666666,0,3,15
 AB,666666,0,5,15
 AC,666666,0,24,15
 AD,666666,0,25,15
 AE,666666,0,26,15
 AF,666666,0,51,15
 AG,666666,0,60,15
 AH,666666,0,73,15
 AI,666666,0,80,15
 AJ,666666,0,99,15

Another annotation marker example demonstrating city altitudes in Switzerland.



[Back to top \(#top\)](#)

All rights reserved. Java is a registered trademark of Oracle and/or its affiliates.

Última actualización: Noviembre 27, 2013.