Google-O-Meter Charts

This document describes how to create a dial-type chart called a Google-o-meter chart.

Table of Contents

Overview

A Google-o-meter is a gauge that points toward a single value on a range. The range has a color gradation that you can optionally specify. You can also specify custom text above the pointer. The dial range is from the data format minimum value to the maximum value. Specify one data value in your `chd` parameter for the arrow to point to.

You can specify multiple arrows on the same chart, and style the arrow body and point size differently on each. If you want the same arrow style for all arrows, specify the data points as values in the same series. If you want to apply different styles to different arrows, group all arrows with the same style into the same series.

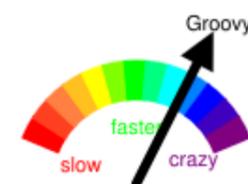Specify a Google-O-Meter chart with the following syntax:

Syntax

```
cht=gom
```

Examples

Description	Example
This demonstrates using the `chxl` to specify custom labels both below and above the dial.	`chxt=x,y` `chxl=0:|Groovy|1:|slow|faster|crazy`
You can specify multiple arrows with multiple data points.	`chd=t:20,40,60` `chdl=1|2|3`
To apply a style to arrows, use the `chls` parameter. Each `chls` description applies to all arrows in the corresponding data series. `chls` has an advanced syntax for Google-o-meters. Each set of values inside a set of pipes describes the corresponding data series. You can optionally add a set of pipe-delimited values that describe the arrowhead size in points, from 0—15, the default being 15. Let's examine the `chls` value in this chart: `chls=3|3,5,5|15|10`	`chd=t:20,40|60` `chls=3|3,5,5|15|10`

- **3** - A single value specifies the line thickness. This applies to all arrows in the first series.
- **3,5,5** - The second series has a thickness, dash length, and space length described.
- **15** - Arrowhead size for arrows in the first series.
- **10** - Arrowhead size for arrow in the second series.

Series Colors `chco`

You can optionally specify the colors of the dial using the `chco` parameter. If not specified, it uses a default gradient.

Syntax

```
chco=<color_1>,...,<color_n>
```

<color>

> Specify two or more colors in <u>RRGGBB hexadecimal format</u> (#gcharts_rgb). Colors are applied in a gradient from left to right, with the first color on the left last color on the right.

Description	Example
This chart shows a simple gradient, from white (FFFFFF) to black (000000)	chco=FFFFFF,000000
This chart shows multiple colors assigned to the dial.	chco=FF0000,FF8040,FFFF00, 00FF00,00FFFF, 0000FF,800080

<u>Back to top</u> (#top)

Google-o-meter Labels `chl`

You can optionally specify a custom label to display above the arrow.

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 <u>here</u> (https://developers.google.com/chart/interactive/docs/querylanguage?hl=es#Setting_the_Query_in_the_Data_Source_URL).

Syntax

```
chl=<label>
```
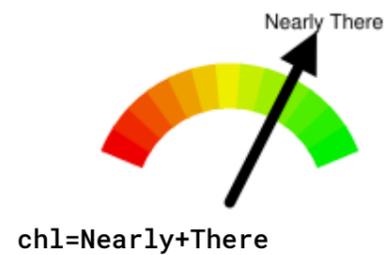
<label>

> Text of the label.

Example

Description	Example
Here's an example of a Google-o-meter with a label.	Nearly There chl=Nearly+There

Back to top (#top)

Standard Features

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

Color Format

Specify colors using a 6-character string of hexadecimal values, plus two optional transparency values, in the format *RRGGBB[AA]*. For example:

- `FF0000` = Red
- `00FF00` = Green
- `0000FF` = Blue
- `000000` = Black
- `FFFFFF` = White

AA is an optional transparency value, where `00` is completely transparent and `FF` is completely opaque. For example:

- `0000FFFF` = Solid blue
- `0000FF66` = Transparent blue

Back to top (#top)

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 `chtt` 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

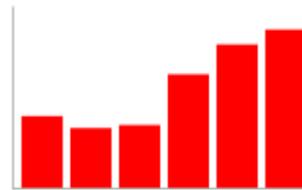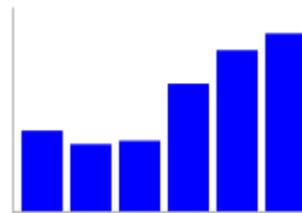| Description	Example

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.
	Site visitors by month
	 January to July
	
	 `chtt=Site+visitors+by+month|`
	 `January+to+July`

A chart with a blue, right-aligned, 20-point title.	Site visitors
	
	 `chtt=Site+visitors`
	 `chts=FF0000,20,r`

Back to top (#top)

Chart Legend Text and Style `chdl`, `chdlp`, `chdls` [*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).

Syntax

```
chdl=<data_series_1_label>|...|<data_series_n_label>
chdlp=<opt_position>|<opt_label_order>
chdls=<color>,<size>
```

`chdl` - 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.

chdlp - [*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 **chdl** to be skipped in the legend. Examples: **chdlp=bv**, **chdlp=r**, **chdlp=bv|r**, **chdlp=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 <u>chma</u> (#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 **chdl**.
- **r** - Display labels in the reverse order as given to **chdl**. 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 **chdl**, separated by commas.

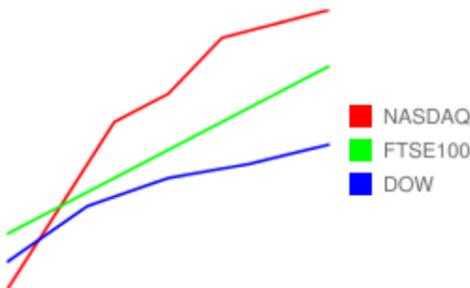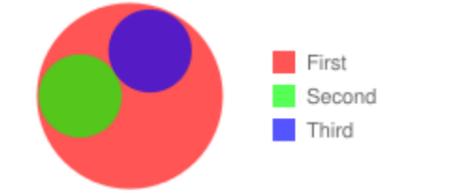**chdls** - [*Optional*] Specifies the color and font size of the legend text.

<color>

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

<size>

The point size of the legend text.

Examples

Description	Example
Two examples of legends. Specify legend text in the same order as your data series.	 **chdl=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)

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:
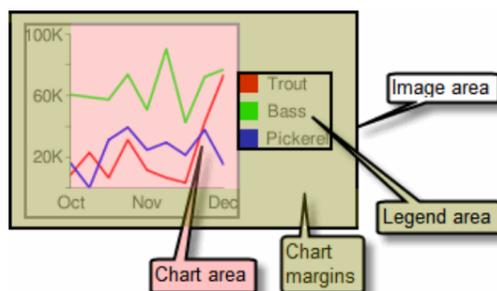
The *chart margins* include the *axis labels* and the *legend area*. The legend area resizes automatically to fit the text exactly, unless you specify a larger width using `chma`, in which case it will expand the margin size wider, squeezing the chart area smaller. You cannot crop a legend by specifying 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 resizeable 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.

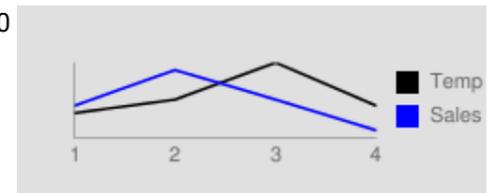**Examples**

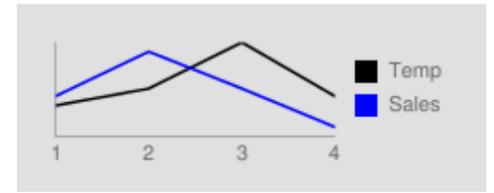| Description	Example
In this example, the chart has a minimum margin of 30 pixels on each side. Because the chart legend is more than 30 pixels wide, the margin on the right side is set to the width of the chart legend, and is different from the other margins. Axis labels are outside the plot area, and are therefore drawn within the margin space.	 `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 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)

Axis Styles and Labels [*Line, Bar, Google-o-meter, Radar, Scatter*]

You can specify which axes to display on the chart, and give them custom labels and positions, ranges, and styles.

Not all charts show axis lines by default. You can specify exactly which axes your chart should show using the `chxt` parameter. Default axis lines do not show numbers; you must specify an axis in the `chxt` parameter to show numbers.

You can choose to have your axes display numbers reflecting the data values, or you can specify custom axes. The default is to display numeric values, with values scaled to range from 0—100. However, you can change that range using `chxr` to display any range, and you can style the values (for example, to show currency symbols or decimal places) using `chxs`.

If you choose to use custom values, for example: "Mon, Tues, Wed", you can use the `chxl` parameter. To place these labels in specific locations along the axis, use the `chxp` parameter.

Finally, you can use the `chxs` and `chxtc` parameters to specify color, size, alignment, and other properties of both custom and numeric axis labels.

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).

This section covers the following topics:

- Visible Axes (#axis_type) (`chxt`) - Which axes to display.
- Axis range (#axis_range) (`chxr`) - Value range for each axis.
- Custom Axis Labels (#axis_labels) (`chxl`) - Custom values to show on the axis.
- Axis label positions (#axis_label_positions) (`chxp`) - Placement of custom labels along each axis.
- Axis label styles (#axis_label_styles) (`chxs`) - Color, size, alignment, and formatting of axis labels.
- Axis tick mark styles (#axis_tick_marks) (`chxtc`) - Length of tick marks for a specific axis.

Visible Axes `chxt`

Bar, line, radar, and scatter charts show one or two axis lines by default, but these lines do not include values. To display values on your axis lines, or to change which axes are shown, you must use the `chxt` parameter. By default, the axis values range from 0-100, unless you scale them explicitly using the `chxr` property. To hide all axis lines in a line chart, specify `:nda` after the chart type value in the `cht` parameter (example: cht=`lc:nda`).

By default, the top and bottom axes do not show tick marks by the values, while the left and right axes do show them. You can change this behavior using the chxs (#axis_label_styles) parameter.

Syntax
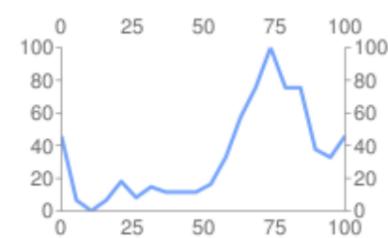
```
chxt=
  <axis_1>
    ,...,
  <axis_n>
```
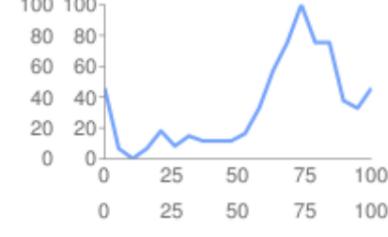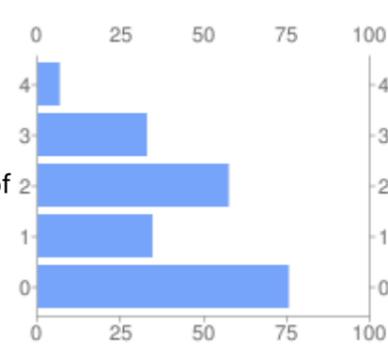
<axis>

An axis to show on the chart. Available axes are:

- `x` - Bottom x-axis
- `t` - Top x-axis [*Not supported by Google-o-Meter*]
- `y` - Left y-axis
- `r` - Right y-axis [*Not supported by Google-o-Meter*]

You can specify multiple axes of the same type, for example: `cht=x,x,y`. This will stack two sets of x-axes along the bottom of the chart. This is useful when adding custom labels along an axis that shows numeric values (see the example below). Axes are drawn from the inside out, so if you have `x,x`, the first x refers to the innermost copy, the next x refers to the next outwards copy, and so on.

Examples

Description	Example
This example shows a line chart with an x-axis, a y-axis, a top axis (t), and a right axis (r). Because no labels are specified, the chart defaults to a range of 0 to 100 for all axes. Note that by default, the top and bottom axes don't show tick marks by the labels.	 `chxt=x,y,r,t`
You can include multiple sets of labels for each axis by including the same value more than once. This example shows two sets of x and two sets of y-axes. This isn't particularly useful when using only the default axis labels, as is shown here. But you can specify custom labels for each copy of each axis, using the `chxl` parameter.	 `chxt=x,x,y,y`

	`chxt=x,x,y,y`
	`chxl=`
	`  1:|Freezing|Hot|`
	`  3:|Low|High`

This example shows a horizontal bar chart with an x-axis, a y-axis, an upper t-axis, and a right r-axis. Axis labels are omitted, so the Chart API displays a range of 0 to 100 for the x-axis and for the t-axis. The range for the y-axis and for the r-axis is determined by the number of bars. In this case, there are five bars, so the Chart API displays a range of 0 to 4. The first label is centered at the base of the first bar, the second label is centered at the base of the second bar, and so on.	 `chxt=x,y,r,t`

You can suppress default axes in a line chart by specifying `:nda` after the chart type.

```
cht=lc:nda
```

Axis Range `chxr`

You can specify the range of values that appear on each axis independently, using the chxr parameter. Note that this does *not* change the scale of the chart elements, only the scale of the axis labels. If you want to make the axis numbers describe the actual data values, set *<start_val>* and *<end_val>* to the lower and upper values of your data format range, respectively. See Axis Scaling (https://developers.google.com/chart/image/docs/data_formats?hl=es#axis_scale) for more information.

You must make an axis visible using the chxt (#axis_type) parameter if you want to specify its range.

To specify custom axis values, use the chxl (#axis_labels) parameter.

Syntax

Separate multiple axis label ranges using the pipe character (|).

```
chxr=
  <axis_index>,<start_val>,<end_val>,<opt_step>
    |...|
  <axis_index>,<start_val>,<end_val>,<opt_step>
```

<axis_index>

> Which axis to apply the labels to. This is a zero-based index into the axis array specified by `chxt`. For example, the r-axis would be 1 in `chxt=x,r,y`.

<start_val>

> A number, defining the low value for this axis.

<end_val>

> A number, defining the high value for this axis.

<opt_step>

> [*Optional*] The count step between ticks on the axis. There is no default step value; the step is calculated to try to show a set of nicely spaced labels.
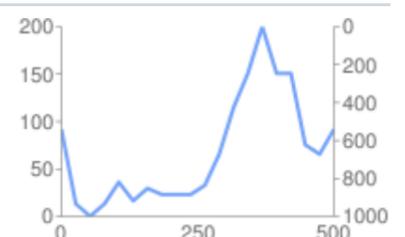
Examples

| Description	Example

This example shows left and right y-axes (**y** and **r**) and one x-axis (**x**).
 	
Each axis has a defined range. Because no labels or positions are specified, values are taken from the given range, and are evenly spaced within that range. In the line chart, values are evenly spread along the x-axis.
 	
Axis direction is reversed for the r-axis (index **2**), because the first value (**1000**) is larger than the last value (**0**).
	
	```chxt=x,y,r```
	```chxr=```
	`  0,0,500|`
	`  1,0,200|`
	`  2,1000,0`

In this example, values are specified for the x-axis.

Axis labels are evenly spaced along the axis. A value of five (**5**) is specified for the *<opt_step>* parameter.

```
chxt=x
chxr=0,10,50,5
```
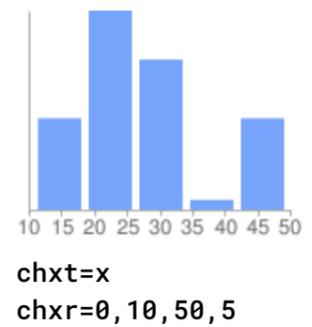
Custom Axis Labels `chxl`

You can specify custom string axis labels on any axis, using the `chxl` parameter. You can specify as many labels as you like. If you display an axis (using the `chxt` parameter) and do not specify custom labels, the standard, numeric labels will be applied. To specify a custom numeric range, use the chxr (#axis_range) parameter (#axis_range) instead.

To set specific locations along the axis for your labels, use the `chxp` parameter.

Syntax

Specify one parameter set for each axis that you want to label. Separate multiple sets of labels using the pipe character (**|**).

```
chxl=
  <axis_index>:|<label_1>|...|<label_n>
    |...|
  <axis_index>:|<label_1>|...|<label_n>
```

<axis_index>

> Which axis to apply labels to. This is an index into the `chxt` parameter array. For example, if you have `chxt=x,x,y,y` then index 0 would be the first x-axis, 1 would be the second x-axis.

<label_1>| ... |<label_n>

> One or more labels to place along this axis. These can be string or number values; strings do not need to be in quotes. *label_1* is displayed at the lowest position on the axis, and *label_n* is displayed at the highest position. Additional labels are spaced evenly between them. Indicate spaces with a + mark. There is no way to specify a line break in a label. Separate labels with a pipe character. **Note:** Do not place a pipe after the final label in the `chxl` parameter.

Examples

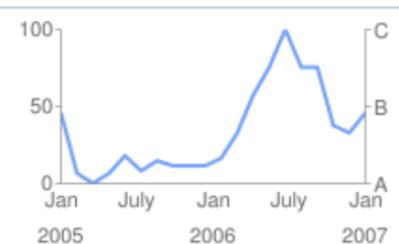| Description	Example

This chart shows how to add custom labels to two axes. Note how the values are evenly spaced, and how the last `chxl` value does not end with a pipe.	`chxt=x,y`
	`chxl=`
	`0:|Jan|Feb|March|April|May|`
	`1:|Min|Mid|Max`

This example includes axis labels on the left and right y-axes (**y** and **r**). It also includes two sets of values for the x-axis (**x**). You could consider adding tick marks on the y-axis using chxs (#axis_label_styles).	`chxt=x,y,r,x`
	`chxl=`
	`0:|Jan|July|Jan|July|Jan|`

```
1:|0|50|100|
2:|A|B|C|
3:|2005|2006|2007
```

This example includes axis labels on the left and right y-axes (**y** and **r**). It also includes two sets of values for the x-axis (**x**). Note the empty labels for the lower x-axis set, used to space the values apart.

This example uses default values for the axis labels on the left y-axis.
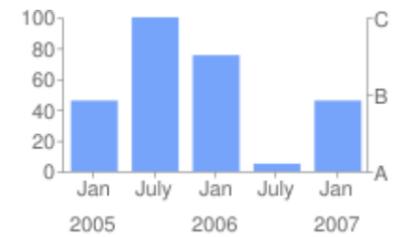
```
chxt=x,y,r,x
chxl=
0:|Jan|July|Jan|July|Jan|
2:|A|B|C|
3:|2005||2006||2007
```

If you want to add a generic label to describe a whole axis (for example, to label one axis "cost" and another "student"), use the **chxt** property to add an additional axis on each side, then use **chxl** to add a single custom label to each side, and **chxp** to space it in the middle of the axis.
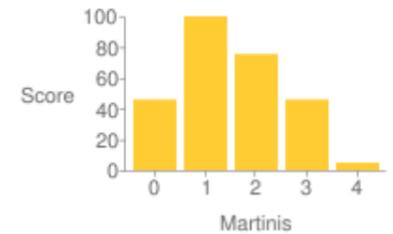
```
chxt=x,x,y,y
chxl=1:|Martinis|3:|Score
chxp=1,50|3,50
```

Back to top (#top)

Axis Label Positions `chxp`

You can specify which axis labels to show, whether using the default labels or custom labels specified using `chxl`. If you do not specify exact positions using this parameter, labels will be spaced evenly and at a default step value along the axes. If you do not specify `chxl`, then the tick mark labels will be the default values (typically data values, or the bar numbers in bar charts).

Syntax

Separate multiple positioning sets using the pipe character (`|`).

```
chxp=
  <axis_1_index>,<label_1_position>,...,<label_n_position>
    |...|
  <axis_m_index>,<label_1_position>,...,<label_n_position>
```

<axis_index>

The axis for which you are specifying positions. This is an index into the `chxt` parameter array. For example, if you have `chxt=x,x,y,y` then index 0 would be the first x-axis, 1 would be the second x-axis, and so on.

<label_1_position>,...,<label_n_position>

The position of the label along the axis. This is a comma-separated list of numeric values, where each value sets the position of the corresponding label in the `chxl` array: the first entry applies to the first label, and so on. The position is a value in the range for that axis (https://developers.google.com/chart/image/docs/data_formats?hl=es#axis_scale). Note that this will always be 0—100 unless you have specified a custom range using `chxr`. You must have as many positions as you have labels for that axis.
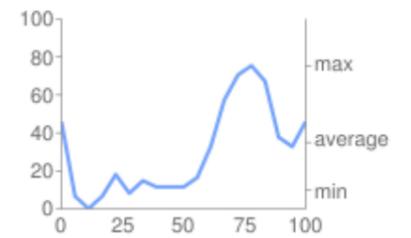
Examples

Description	Example

This example includes r-axis labels at specified positions on the chart. The label text is specified using the `chxl` parameter.

Labels with a specified position of **0** are placed at the bottom of the y- or r-axis, or at the left of the x- or t-axis.

Labels with a specified position of **100** are placed at the top of the y- or r-axis, or at the right of the x- or t-axis.

```
chxt=x,y,r
chxl=2:|min|average|max
chxp=2,10,35,75
```

This example demonstrates showing the default label values, but only at specified locations.

chxp=1,10,35,75 - The y-axis should show only three labels: 10, 35, and 75. Because no custom label text is specified, these axis values are shown. Note how you don't have to space labels evenly apart when you use **chxp**. If **chxp** had not been specified here, the default label value distance on the y-axis would be every 20 units, as shown in the second graph.
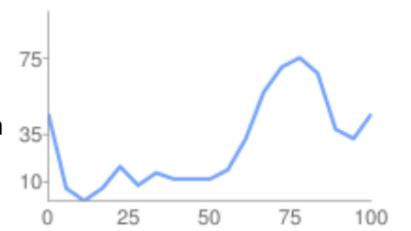

```
chxt=x,y
chxp=1,10,35,75
```
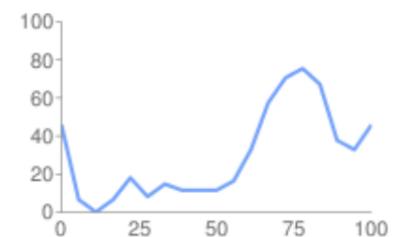


```
chxt=x,y
```
chxp not specified

Axis Label Styles `chxs`

You can specify the font size, color, and alignment for axis labels, both custom labels and default label values. All labels on the same axis have the same format. If you have multiple copies of an axis, you can format each one differently. You can also specify the format of a label string, for example to show currency symbols or trailing zeroes.

By default, the top and bottom axes do not show tick marks by the values, while the left and right axes do show them.

Syntax

Values for multiple axes should be separated using a pipe character (|).

```
chxs=
 <axis_index><opt_format_string>,<opt_label_color>,<opt_font_size>,<opt_alignment>,<opt_axis_or_tick>,<opt_tick_colo
   |...|
 <axis_index><opt_format_string>,<opt_label_color>,<opt_font_size>,<opt_alignment>,<opt_axis_or_tick>,<opt_tick_colo
```

<axis_index>

The axis to which this applies. This is a zero-based index into the `chxt` parameter.

<opt_format_string>

[*Optional*] This is an optional format string that, if used, follows immediately after the axis index number without an intervening comma. It starts with a literal letter N followed by the following values, all optional: The formatting string syntax is as follows:
N*<preceding_text>*<*number_type*><*decimal_places*>zs<*x or y*>*<*following_text*>
Here is the meaning of each element:

- *<preceding_text>* - Literal 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), 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*>** - Literal text to follow each value.

<opt_label_color>

The color to apply to the axis text (but not axis line), in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). Axis line color is specified separately using *opt_axis_color. Default is gray.*

<opt_font_size>

[*Optional*] specifies the font size in pixels. This parameter is optional.

<opt_alignment>

[*Optional*] Label alignment. For top or bottom axes, this describes how the label aligns to the tick mark above or below it; for left or right axes, this describes how the aligns inside its bounding box, which touches the axis. Specify one of the following numbers:

- **–1** - *Top or bottom*: labels are to the right of the ticks; *Left or right*: labels are left-aligned in their area. *Default for r-axis labels.*

- **0** - *Top or bottom*: labels are centered on the ticks; *Left or right*: labels are centered in their area. *Default for x- and t-axis labels.*

- **1** - *Top or bottom*: labels are to the left of the ticks; *Left or right*: labels are right-aligned in their area. *Default for y-axis labels.*

<opt_axis_or_tick>

[*Optional; not supported in Google-o-meter*] Whether to show tick marks and/or axis lines for this axis. Tick marks and axis lines are only available for innermost axes (for example, they are not supported for the outer of two x-axes). Use one of the following values:

- **l** (lowercase 'L') - Draw axis line only.

- **t** - Draw tick marks only. Tick marks are the little lines next to axis labels.

- **lt** - [*Default*] Draw both an axis line and tick marks for all labels.

- **_** - (Underscore) Draw neither axis line nor tick marks. If you want to hide an axis line, use this value.

<tick_color>

[*Optional; not supported in Google-o-meter*] The tick mark color, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). *Default is gray.*

<opt_axis_color>

[*Optional*] The color of this axis line, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). *Default is gray.*
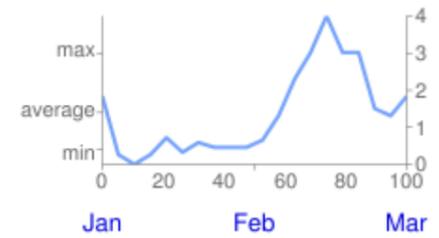
Examples

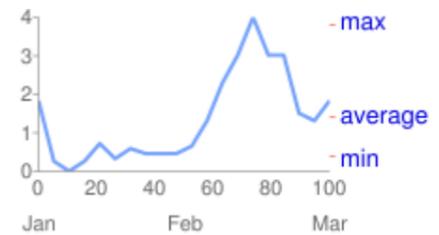| Description	Example
Font size and color are specified for the second x-axis (Jan, Feb, Mar).	

```
chxt=x,y,r,x
chxr=2,0,4
chxl=3:|Jan|Feb|Mar|
     1:|min|average|max
chxp=1,10,35,75
chxs=3,0000DD,13,0,t
```

Font size, color, and alignment are specified for the right y-axis. Tick marks, but no axis line, are drawn.



```
chxt=x,y,r,x
chxl=3:|Jan|Feb|Mar|
     2:|min|average|max
chxp=2,10,35,95
chxs=2,0000DD,13,-1,t,FF0000
```

This chart includes three data sets, and shows three sets of axis labels, one per series. Each set of labels is formatted using a custom formatting string, as described here:

- `0N*e,000000|`
  - `0` means the first data series
  - `N` means a formatting string
  - `*` means the start of the format specifiers
  - `e` means scientific notation
  - `*` means the end of the format specifiers
  - `000000` means black text.
- `1N*cUSD*Mil,FF0000|`
  - `1` means the second series
  - `N` means a formatting string
  - `*` means the start of the format specifiers
  - `c` means a currency marker
  - `USD` specifies the US dollar as the currency marker to use
  - `*` means the end of the format specifiers
  - `Mil` is a literal following string
  - `FF0000` means red text.
- `2N*sz2*,0000FF`
  - `2` means the third series
  - `N` means a formatting string
  - `*` means the start of the format specifiers
  - `s` means to show grouping specifiers (in US English locale, that is a comma every three zeroes)
  - `z2` means show two trailing zeroes
  - `0000FF` means blue text.
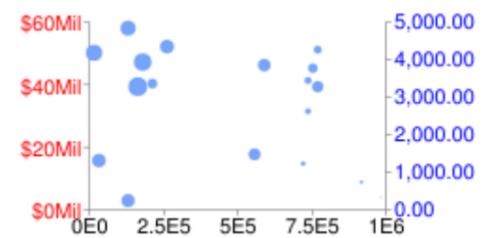

```
chd=s:
  984sttvuvkQIBLKNCAIi,
  DEJPgq0uov17zwopQODS,
  AFLPTXaflptx159gsDrn
chxr=
  0,0,1000000,250000|
  1,0,60|
  2,0,5000
chxs=
  0N*e,000000|
  1N*cUSD*Mil,FF0000|
  2N*sz2*,0000FF
```

The axis label ranges are set using the `chxr` parameter (*axis_index*, *start*, *end*, *step*). If not set, they would have been 0—100 by default.

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

Axis Tick Mark Styles `chxtc`

You can specify long tick marks for specific axes. Typically this is used to extend a tick mark across the length of a chart. Use the `chxs` parameter to change the tick mark color.

Values for multiple axes should be separated using a pipe character (`|`). Values within a series should be separated by a comma.

Syntax

```
chxtc=
  <axis_index_1>,<tick_length_1>,...,<tick_length_n>
    |...|
  <axis_index_m>,<tick_length_1>,...,<tick_length_n>
```

<axis_index>

> The axis to which this applies. This is a zero-based index into the `chxt` parameter. Separate values for different axes using a bar delimiter.

<tick_length_1>,…,<tick_length_n>

> Length of the tick marks on that axis, in pixels. If a single value is given, it will apply to all values; if more than one value is given, the axis tick marks will cycle through the list of values for that axis. Positive values are drawn outside the chart area and cropped by the chart borders. The maximum positive value is 25. Negative values are drawn inside the chart area and cropped by the chart area borders.

Examples

| Description	Example

Example of using `chxtc` to create long red tick marks. The tick mark length here exceeds the chart area width, but is cropped to fit within the chart.
 	
 • `chxt=x,y,r,x` - Show a left axis, a right axis, and two bottom axes.
 • `chxl=2:|min|average|max|3:|Jan|Feb|Mar` - The label text assigned to the 'r' (right side) and outer x-axes.
 • `chxp=2,10,35,95` - Custom label positions along the r-axis (index=2) for the three labels.
 • `chxs=2,0000dd,13,-1,t,FF0000` - Axis label styles for the r-axis: text color, text size, left-aligned, with red tick marks.
 • `chxtc=1,10|2,-180` - Axis tick lengths for the y- and r-axes. The first value specifies 10-pixel-long ticks, outside the axis. The second value specifies 180-pixel-long ticks inside the axis; the negative number means that the tick goes inside the axis, and the tick is cropped to fit inside the chart.
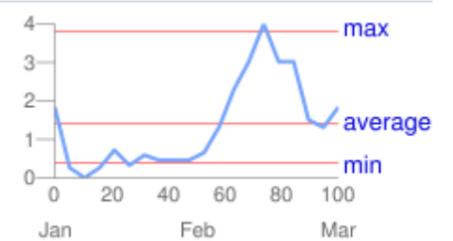	
	 `chxt=x,y,r,x`
	 `chxl=`
	 `  2:|min|average|max|`
	 `  3:|Jan|Feb|Mar`
	 `chxp=2,10,35,95`
	 `chxs=`
	 `  2,0000dd,13,-1,t,FF0000`
	 `chxtc=1,10|2,-180`

This chart demonstrates alternating tick lengths. `chxtc` specifies two tick length values for the y-axis (5 and 15), and the ticks drawn on the chart alternate between the two values.	
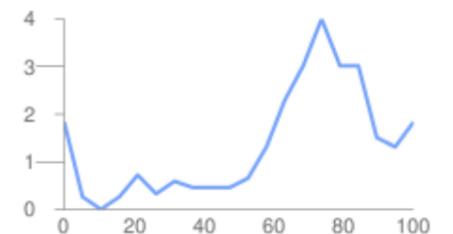	
	 `chxt=x,y`
	 `chxtc=`
	 `  1,5,15`

Back to top (#top)

Line Styles `chls` [*Line, Radar*]

You can specify line thickness and solid/dashed style with the `chls` parameter. This parameter can only be used to style lines in line or radar charts; you cannot use it to style the line in a compound chart lines, unless the base type of the compound chart is a line chart.

Syntax

Separate multiple line styles by the pipe character (|); the first style applies to the first line, the second to the next, and so on. If you have fewer styles than lines, the default style is applied to all the unspecified lines.

```
chls=
  <line_1_thickness>,<opt_dash_length>,<opt_space_length>
    |...|
  <line_n_thickness>,<opt_dash_length>,<opt_space_length>
```
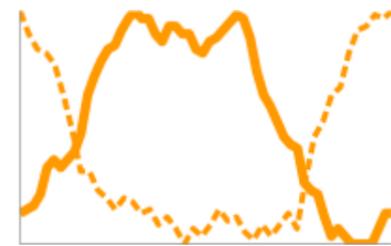
<line_1_thickness>

> Thickness of the line, in pixels.

<opt_dash_length>, <opt_space_length>

> [*Optional*] Used to define dashed grid lines. The first parameter is the length of each line dash, in pixels. The second parameter is the spacing between dashes, in pixels. For a solid line, specify neither value. If you only specify *<opt_dash_length>*, then *<opt_space_length>* will be set to *<opt_dash_length>*. *Default is 1,0 (a solid line)*.

Examples

Description	Example
Here the dashed line is specified by `3,6,3` and the thicker, solid line is specified by `5`..	`chls=3,6,3|5`

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) for an example.

s
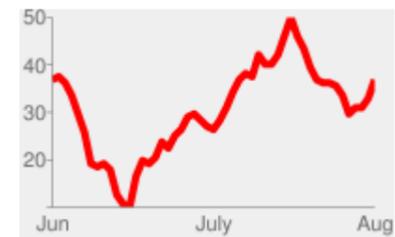
> Indicates a solid or transparency fill.

<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.
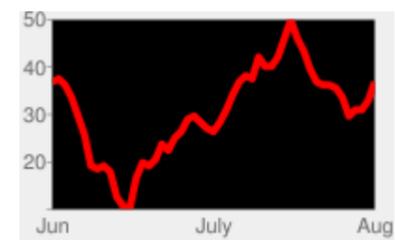
Examples

Description	Example
This example fills the chart background with pale gray (**EFEFEF**).	`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) (#top)

Gradient Fills `chf` [*Line, Bar, Google-o-meter, Radar, Scatter,Venn*]

You can apply one or more gradient fills to chart areas or backgrounds. Gradient fills are fades from a one color to another color. (Pie, Google-o-meter charts: background only.)

Each gradient fill specifies an angle, and then two or more colors anchored to a specified location. The color varies as it moves from one anchor to another. You must have at least two colors with different *<color_centerpoint>* values, so that one can fade into the other. Each additional gradient is specified by a *<color>*,*<color_centerpoint>* pair.

Syntax

```
chf=<fill_type>,lg,<angle>,<color_1>,<color_centerpoint_1>
    ,...,
  <color_n>,<color_centerpoint_n>
```

<fill_type>

> The chart area to fill. One of the following:
> - `bg` - Background fill
> - `c` - Chart area fill.

- **b***<index>* - Bar gradient fills (bar charts only). Replace *<index>* with the series index of the bars to fill with a gradient. See Bar Chart Series Colors (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) for an example.

lg

Specifies a gradient fill.

<angle>

A number specifying the angle of the gradient from 0 (horizontal) to 90 (vertical).

<color>

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

<color_centerpoint>

Specifies the anchor point for the color. The color will start to fade from this point as it approaches another anchor. The value range is from 0.0 (bottom or left edge) to 1.0 (top or right edge), tilted at the angle specified by *<angle>*.
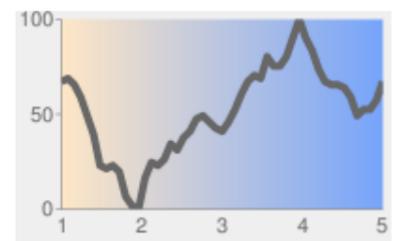
Examples

| Description	Example

Chart area has a horizontal linear gradient, specified with an angle of zero degrees (`0`).
 	
The colors are peach (`FFE7C6`), centered on the left side (position `0.0`) and blue (`76A4FB`) centered on the right side (position `1.0`).
 	
The chart background is drawn in gray (`EFEFEF`).
	
	`chf=`
	`  c,lg,0,`
	`  FFE7C6,0,` (peach)
	`  76A4FB,1` (blue)

Chart area has a diagonal (bottom left to top right) linear gradient, specified with an angle of forty-five degrees (`45`).
 	
Peach (`FFE7C6`) is the first color specified. The bottom left of the chart is pure peach.
 	
Blue (`6A4FB`) is the second color specified. The top right of the chart is pure blue. Note how we specify an offset of 0.75, to provide a peak of blue that fades away towards the top right corner.
 	
The chart background is drawn in gray (`EFEFEF`).
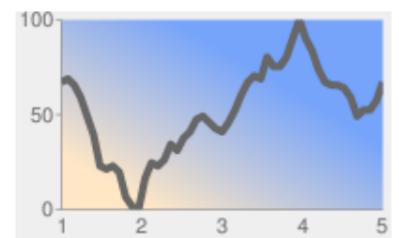	
	`chf=`
	`  c,lg,45,`
	`  FFE7C6,0,` (peach)
	`  76A4FB,0.75` (blue)

Chart area has a vertical (top to bottom) linear gradient, specified with an angle of ninety degrees (`90`).
 	
Blue (`76A4FB`) is the first color specified. The top of the chart is pure blue.
 	
Peach (`FFE7C6`) is the second color specified. The bottom of the chart is pure peach.
 	
The chart background is drawn in gray (`EFEFEF`).
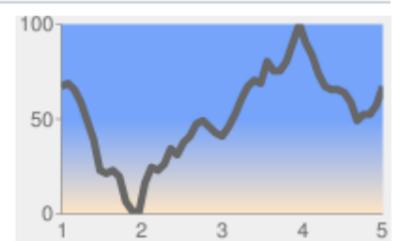	
	`chf=`
	`  c,lg,90,`
	`  FFE7C6,0,` (peach)
	`  76A4FB,0.5` (blue)

Back to top (#top)

Striped fills `chf` [*Line, Bar, Google-o-meter, Radar, Scatter, Venn*]

You can specify a striped background fill for your chart area, or the whole chart. (Pie, Google-o-meter charts: background only.)

Syntax

```
chf=
  <fill_type>,ls,<angle>,<color_1>,<width_1>
    ,...,
  <color_n>,<width_n>
```

<fill_type>

> The chart area to fill. One of the following:
> - **bg** - Background fill
> - **c** - Chart area fill
> - **b*<index>*** - Bar striped fills (bar charts only). Replace *<index>* with the series index of the bars to fill with stripes. See Bar Chart Series Colors (https://developers.google.com/chart/image/docs/gallery/bar_charts?hl=es) for an example.

ls

> Specifies linear stripe fill.

<angle>

> The angle of all stripes, relative to the y-axis. Use `0` for vertical stripes or `90` for horizontal stripes.

<color>

> The color for this stripe, in RRGGBB hexadecimal format (https://developers.google.com/chart/image/docs/chart_params?hl=es#gcharts_rgb). Repeat *<color>* and *<width>* for each additional stripe. **You must have at least two stripes.** Stripes alternate until the chart is filled.
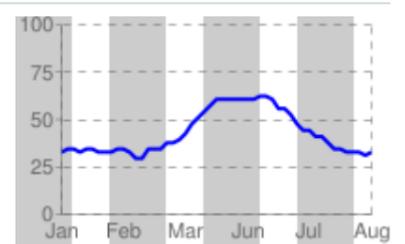
<width>

> The width of this stripe, from `0` to `1`, where `1` is the full width of the chart. Stripes are repeated until the chart is filled. Repeat *<color>* and *<width>* for each additional stripe. **You must have at least two stripes.** Stripes alternate until the chart is filled.
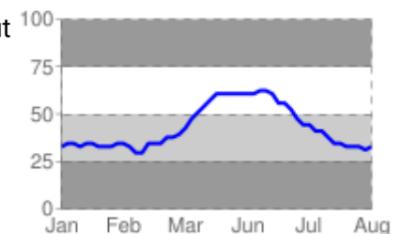
Examples

Description	Example

bg,ls,0 - Background stripe fill with stripes at a zero degree angle to the y-axis (parallel to the y-axis). The stripes fill the chart background as well as the plot area. **CCCCCC,0.15** - The first stripe is dark gray, 15% as wide as the chart. **FFFFFF,0.1** - The second stripe is white, 10% as wide as the chart.	
	``chf=
	  bg,ls,0,
	  CCCCCC,0.15,
	  FFFFFF,0.1```

c,ls,90 - Chart area with horizontal stripes at an angle of ninety degrees from the y-axis. The stripes fill the plot area, but the chart background is omitted. **999999,0.25** - The first stripe is dark gray, 25% as wide as the chart. **CCCCCC,0.25** - Same as the first stripe, but a lighter gray. **FFFFFF,0.25** - Same as the first stripe, but white.	
	``chf=
	  c,ls,90,
	  999999,0.25,
	  CCCCCC,0.25,
	  FFFFFF,0.25```

Back to top (#top)

Última actualización: Noviembre 27, 2013.