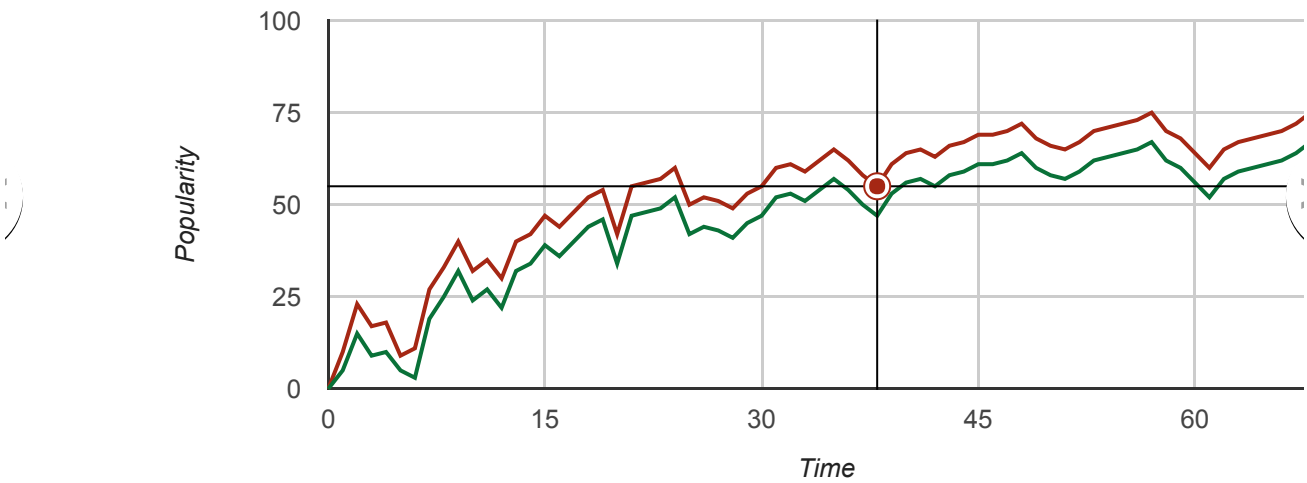
Line Chart

Overview

A line chart that is rendered within the browser using SVG (http://www.w3.org/Graphics/SVG/) or VML (http://en.wikipedia.org/wiki/Vector_Markup_Language). Displays tooltips when hovering over points.

Examples

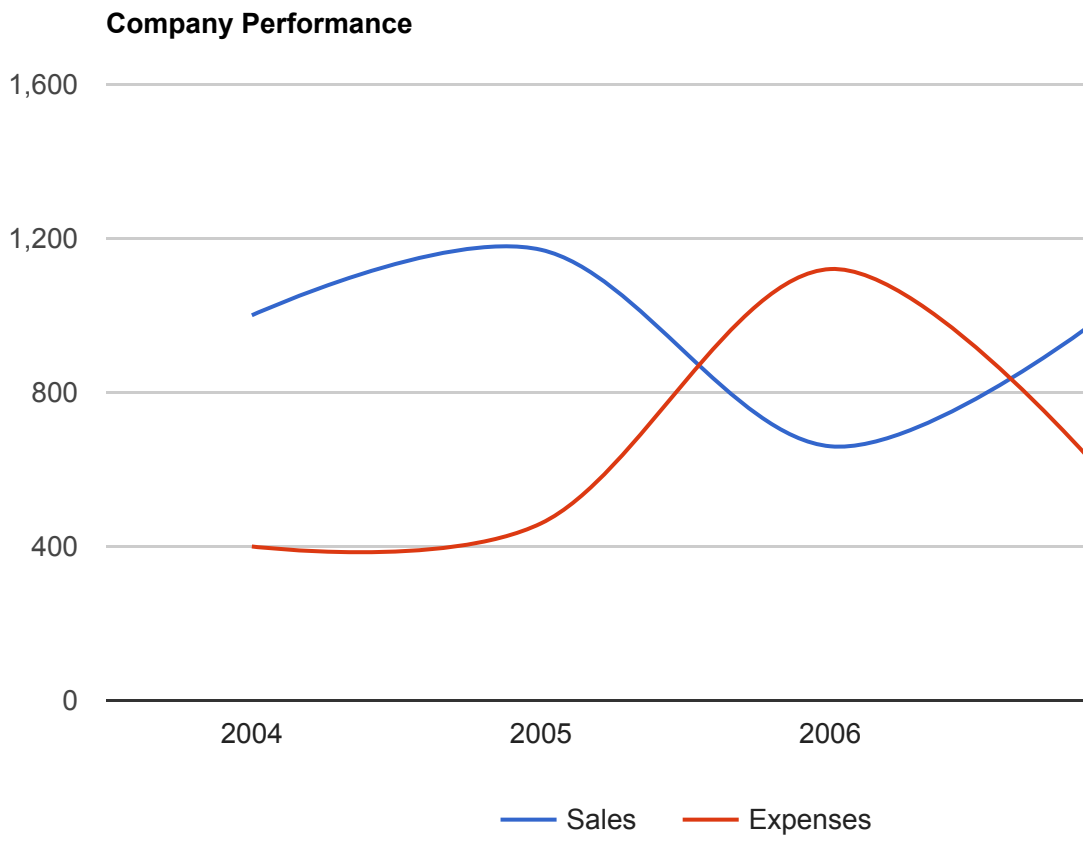
Crosshairs



CODE IT YOURSELF ON JSFIDDLE

Curving the Lines

You can smooth the lines by setting the `curveType` option to `function`:



The code to generate this chart is below. Note the use of the `curveType: function` option:

```
<html>
<head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Year', 'Sales', 'Expenses'],
        ['2004',  1000,      400],
        ['2005',  1170,      460],
        ['2006',  660,       1120],
        ['2007',  1030,      540]
      ]);

      var options = {
        title: 'Company Performance',
        curveType: 'function',
        legend: { position: 'bottom' }
```

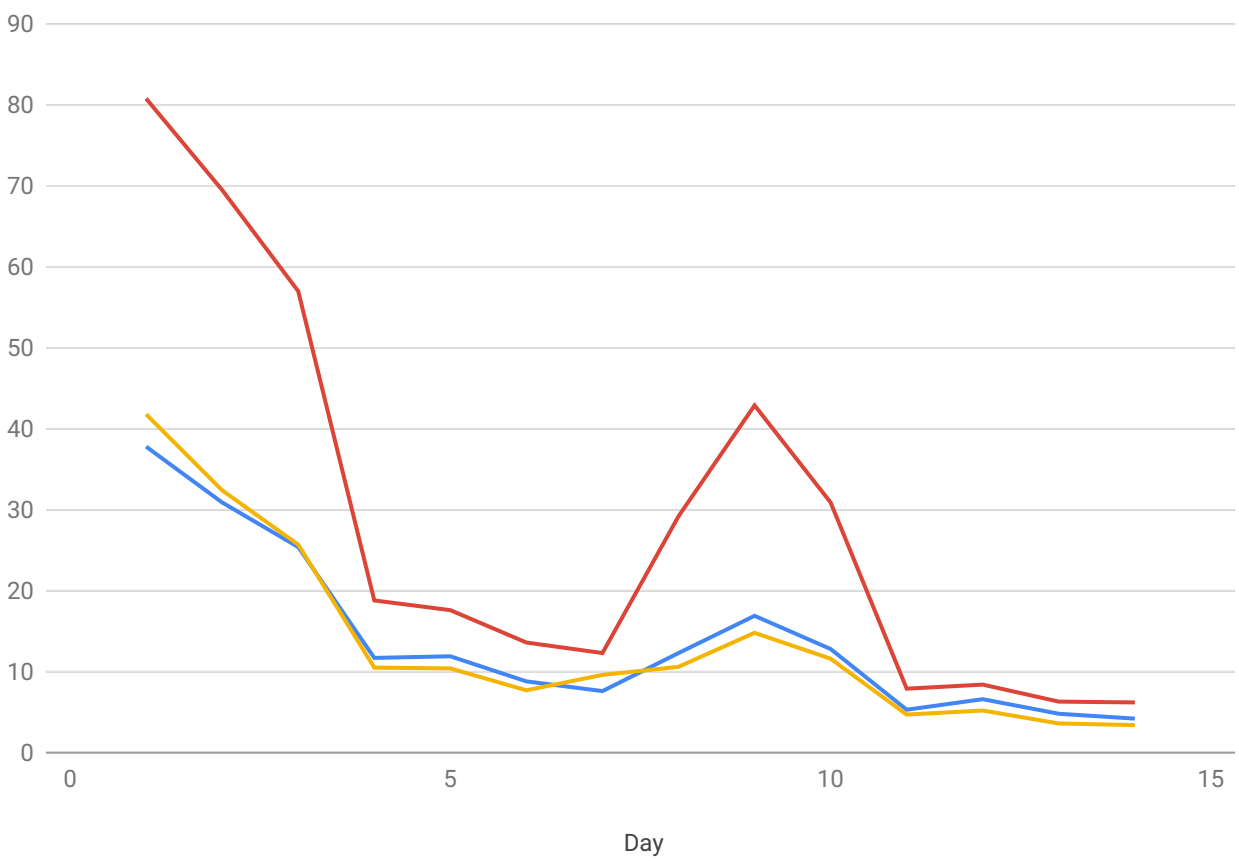
```
        };

        var chart = new google.visualization.LineChart(document.getElementById
        chart.draw(data, options);
      }
    </script>
  </head>
  <body>
    <div id="curve_chart" style="width: 900px; height: 500px"></div>
  </body>
</html>
```

Creating Material Line Charts

In 2014, Google announced guidelines intended to support a common look and feel across its properties and apps (such as Android apps) that run on Google platforms. We call this effort *Material Design*. We'll be providing "Material" versions of all our core charts; you're welcome to use them if you like how they look.

Creating a Material Line Chart is similar to creating what we'll now call a "Classic" Line Chart. You load the Google Visualization API (although with the `'line'` package instead of the `'corechart'` package), define your datatable, and then create an object (but of class `google.charts.Line` instead of `google.visualization.LineChart`).

Note: Material Charts will not work in old versions of Internet Explorer. (IE8 and earlier versions don't support SVG, which Material Charts require.)



Material Line Charts have many small improvements over Classic Line Charts, including an improved color palette, rounded corners, clearer label formatting, tighter default spacing between series, softer gridlines, and titles (and the addition of subtitles).

```
    google.charts.load('current', {'packages':['line']});
    google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = new google.visualization.DataTable();
    data.addColumn('number', 'Day');
    data.addColumn('number', 'Guardians of the Galaxy');
    data.addColumn('number', 'The Avengers');
    data.addColumn('number', 'Transformers: Age of Extinction');

    data.addRows([
      [1,  37.8, 80.8, 41.8],
      [2,  30.9, 69.5, 32.4],
      [3,  25.4,   57, 25.7],
```

```
        [4,  11.7, 18.8, 10.5],
        [5,  11.9, 17.6, 10.4],
        [6,   8.8, 13.6,  7.7],
        [7,   7.6, 12.3,  9.6],
        [8,  12.3, 29.2, 10.6],
        [9,  16.9, 42.9, 14.8],
        [10, 12.8, 30.9, 11.6],
        [11,  5.3,  7.9,  4.7],
        [12,  6.6,  8.4,  5.2],
        [13,  4.8,  6.3,  3.6],
        [14,  4.2,  6.2,  3.4]
      ]);

      var options = {
        chart: {
          title: 'Box Office Earnings in First Two Weeks of Opening',
          subtitle: 'in millions of dollars (USD)'
        },
        width: 900,
        height: 500
      };

      var chart = new google.charts.Line(document.getElementById('linechart_ma

      chart.draw(data, options);
    }
```

The Material Charts are in **beta**. The appearance and interactivity are largely final, but many of the options available in Classic Charts are not yet available in them. You can find a list of options that are not yet supported in this issue (https://github.com/google/google-visualization-issues/issues/2143).

Also, the way options are declared is not finalized, so you must convert your options by replacing this line:

```
chart.draw(data, options);
```

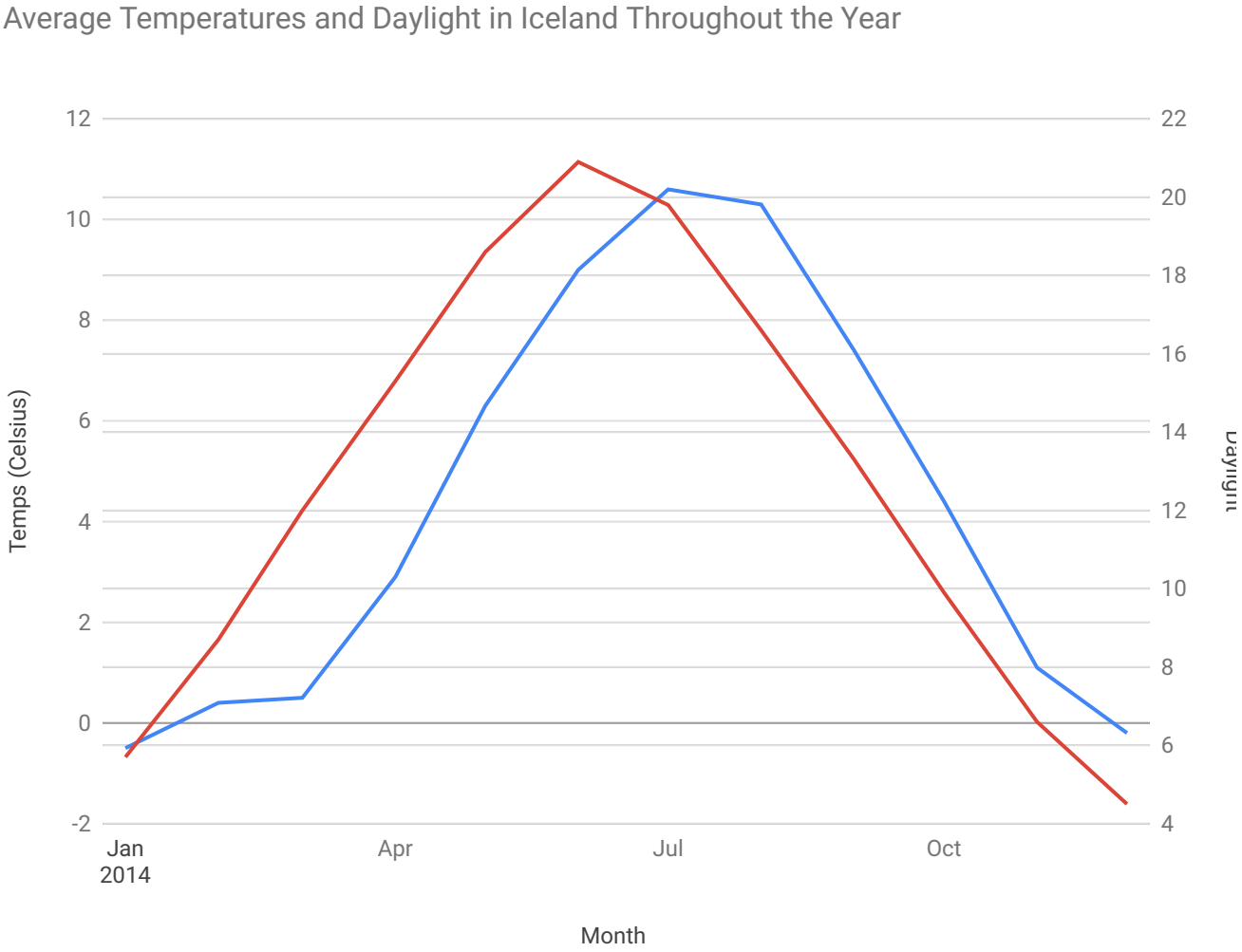
...with this:

```
chart.draw(data, google.charts.Line.convertOptions(options));
```

Dual-Y Charts

Sometimes you'll want to display two series in a line chart, with two independent y-axes: a left axis for one series, and a right axis for another:

CHANGE TO CLASSIC



CODE IT YOURSELF ON JSFIDDLE

Note that not only are our two y-axes labeled differently ("Temps" versus "Daylight") but they each have their own independent scales and gridlines. If you want to customize this behavior, use the `vAxis.gridlines` options.

In the Material code below, the `axes` and `series` options together specify the dual-Y appearance of the chart. The `series` option specifies which axis to use for each (`'Temps'` and `'Daylight'`; they needn't have any relation to the column names in the datatable). The `axes` option then makes this chart a dual-Y chart, placing the `'Temps'` axis on the left and the `'Daylight'` axis on the right.

In the Classic code, this differs slightly. Rather than the `axes` option, you will use the `vAxes` option (or `hAxes` on horizontally oriented charts). Also, instead of using names, you will use

the index numbers to coordinate a series with an axis using the `targetAxisIndex` option.

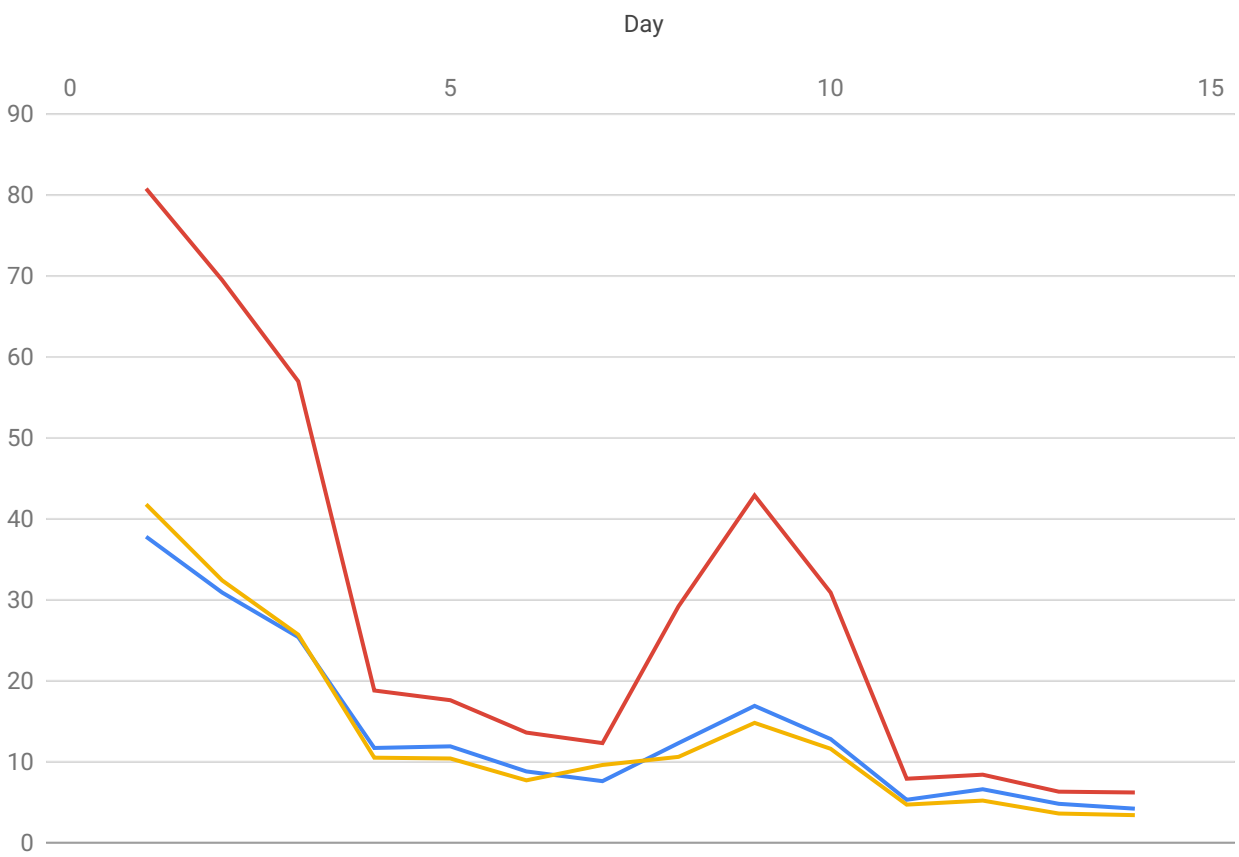
MATERIAL CLASSIC

```
var classicOptions = {
  title: 'Average Temperatures and Daylight in Iceland Throughout the
  width: 900,
  height: 500,
  // Gives each series an axis that matches the vAxes number below.
  series: {
    0: {targetAxisIndex: 0},
    1: {targetAxisIndex: 1}
  },
  vAxes: {
    // Adds titles to each axis.
    0: {title: 'Temps (Celsius)'},
    1: {title: 'Daylight'}
  },
  hAxis: {
    ticks: [new Date(2014, 0), new Date(2014, 1), new Date(2014, 2), n
            new Date(2014, 4),  new Date(2014, 5), new Date(2014, 6),
            new Date(2014, 8), new Date(2014, 9), new Date(2014, 10),
           ]
  },
  vAxis: {
    viewWindow: {
      max: 30
    }
  }
};
```

Top-X Charts

Note: Top-X axes are available only for Material charts (i.e., those with package `line`).

If you want to put the X-axis labels and title on the top of your chart rather than the bottom, you can do that in Material charts with the `axes.x` option:




```
<html>
<head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.j
    <script type="text/javascript">
      google.charts.load('current', {'packages':['line']});
      google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

      var data = new google.visualization.DataTable();
      data.addColumn('number', 'Day');
      data.addColumn('number', 'Guardians of the Galaxy');
      data.addColumn('number', 'The Avengers');
      data.addColumn('number', 'Transformers: Age of Extinction');

      data.addRows([
        [1,  37.8, 80.8, 41.8],
        [2,  30.9, 69.5, 32.4],
        [3,  25.4,   57, 25.7],
        [4,  11.7, 18.8, 10.5],
        [5,  11.9, 17.6, 10.4],
        [6,   8.8, 13.6,  7.7],
```

```
        [7,   7.6, 12.3,  9.6],
        [8,  12.3, 29.2, 10.6],
        [9,  16.9, 42.9, 14.8],
        [10, 12.8, 30.9, 11.6],
        [11,  5.3,  7.9,  4.7],
        [12,  6.6,  8.4,  5.2],
        [13,  4.8,  6.3,  3.6],
        [14,  4.2,  6.2,  3.4]
      ]);

      var options = {
        chart: {
          title: 'Box Office Earnings in First Two Weeks of Opening',
          subtitle: 'in millions of dollars (USD)'
        },
        width: 900,
        height: 500,
        axes: {
          x: {
            0: {side: 'top'}
          }
        }
      };

      var chart = new google.charts.Line(document.getElementById('line_top_x'

      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="line_top_x"></div>
</body>
</html>
```

Loading

The **`google.charts.load`** package name is **`"corechart"`**.

```
google.charts.load("current", {packages: ["corechart"]});
```

For Material Line Charts, the **`google.charts.load`** package name is **`"line"`**.

```
google.charts.load("current", {packages: ["line"]});
```

The visualization's class name is `google.visualization.LineChart`.

```
var visualization = new google.visualization.LineChart(container);
```

For Material Line Charts, the visualization's class name is `google.charts.Line`.

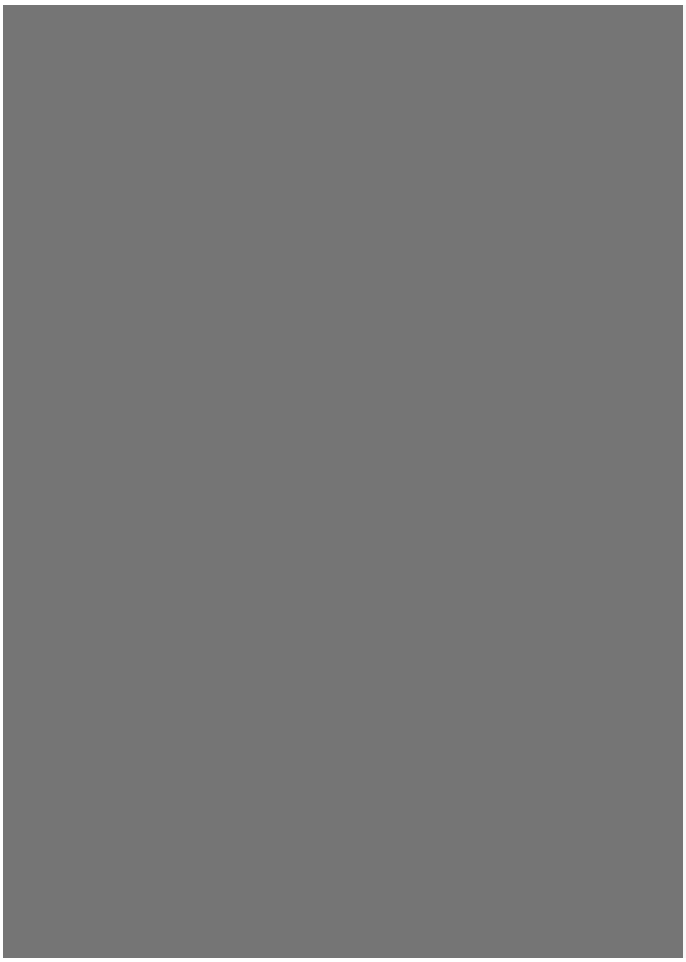
```
var visualization = new google.charts.Line(container);
```

Data Format

Rows: Each row in the table represents a set of data points with the same x-axis location.

Columns:

	Column 0
Purpose:	• X-axis group labels (discrete (https://google-developers.appspot.com/chart/interactive/doc) • X-axis values (continuous (https://google-developers.appspot.com/chart/interactive/doc)
Data Type:	• string (discrete (https://google-developers.appspot.com/chart/interactive/doc) • number, date, datetime or timeofday (continuous (https://google-developers.appspot.com/chart/interactive/doc)
Role:	domain
Optional column roles (https://google-developers.appspot.com/chart/interactive/docs/roles) :	• annotation (https://google-developers.appspot.com/chart • annotationText (https://google-developers.appspot.com/chart



Configuration Options

Name	
aggregationTarget	How multiple data selections are rolled up into tooltips: • `'category'`: Group selected data by x-value. • `'series'`: Group selected data by series. • `'auto'`: Group selected data by x-value if all selections have otherwise. • `'none'`: Show only one tooltip per selection. **`aggregationTarget`** will often be used in tandem with **`selec`** **`tooltip.trigger`**, e.g.:
	`var options = {` `  // Allow multiple` `  // simultaneous selections.`

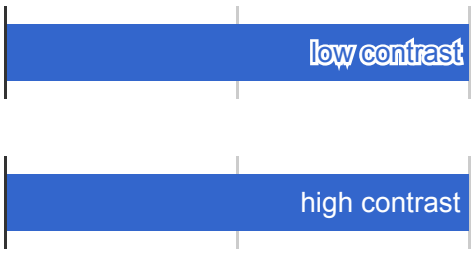
	`selectionMode: 'multiple',` `// Trigger tooltips` `// on selections.` `tooltip: {trigger: 'selection'},` `// Group selections` `// by x-value.` `aggregationTarget: 'category',` `};` **Type:** string **Default:** 'auto'
animation.duration	The duration of the animation, in milliseconds. For details, see t (https://google-developers.appspot.com/chart/interactive/doc **Type:** number **Default:** 0
animation.startup	Determines if the chart will animate on the initial draw. If **true**, baseline and animate to its final state. **Type:** boolean **Default** false
animation.easing	The easing function applied to the animation. The following opt • 'linear' - Constant speed. • 'in' - Ease in - Start slow and speed up. • 'out' - Ease out - Start fast and slow down. • 'inAndOut' - Ease in and out - Start slow, speed up, then slow **Type:** string **Default:** 'linear'
annotations.boxStyle	For charts that support annotations (https://developers.google.com/chart/interactive/docs/roles), object controls the appearance of the boxes surrounding annot `var options = {` `annotations: {` `boxStyle: {` `// Color of the box outline.` `stroke: '#888',` `// Thickness of the box outline.` `strokeWidth: 1,` `// x-radius of the corner curvature.` `rx: 10,` `// y-radius of the corner curvature.`

```
      ry: 10,
      // Attributes for linear gradient fill.
      gradient: {
        // Start color for gradient.
        color1: '#fbf6a7',
        // Finish color for gradient.
        color2: '#33b679',
        // Where on the boundary to start and
        // end the color1/color2 gradient,
        // relative to the upper left corner
        // of the boundary.
        x1: '0%', y1: '0%',
        x2: '100%', y2: '100%',
        // If true, the boundary for x1,
        // y1, x2, and y2 is the box. If
        // false, it's the entire chart.
        useObjectBoundingBoxUnits: true
      }
    }
  }
};
```

This option is currently supported for area, bar, column, combo, supported by the Annotation Chart (https://google-developers.appspot.com/chart/interactive/doc

Type: object
Default: null

annotations.datum	For charts that support annotations (https://google-developers.appspot.com/chart/interactive/doc **annotations.datum** object lets you override Google Charts' for individual data elements (such as values displayed with eac control the color with **annotations.datum.stem.color**, th **annotations.datum.stem.length**, and the style with **anno** **Type:** object **Default:** color is "black"; length is 12; style is "point".
annotations.domain	For charts that support annotations

	(https://google-developers.appspot.com/chart/interactive/doc **`annotations.domain`** object lets you override Google Charts' for a domain (the major axis of the chart, such as the X axis on control the color with **`annotations.domain.stem.color`**, th **`annotations.domain.stem.length`**, and the style with **anr** **Type:** object **Default:** color is "black"; length is 5; style is "point".
annotations.highContrast	For charts that support annotations (https://developers.google.com/chart/interactive/docs/roles), **`annotations.highContrast`** boolean lets you override Goo annotation color. By default, **`annotations.highContrast`** is select an annotation color with good contrast: light colors on da light. If you set **`annotations.highContrast`** to false and do color, Google Charts will use the default series color for the ann  **Type:** boolean **Default:** true
annotations.stem	For charts that support annotations (https://google-developers.appspot.com/chart/interactive/doc **`annotations.stem`** object lets you override Google Charts' ch control color with **`annotations.stem.color`** and the stem le **`annotations.stem.length`**. Note that the stem length optio with style **`'line'`**: for **`'line'`** datum annotations, the stem ler text, and for **`'line'`** domain annotations, the stem extends acr **Type:** object **Default:** color is "black"; length is 5 for domain annotations and
annotations.style	For charts that support annotations (https://google-developers.appspot.com/chart/interactive/doc **`annotations.style`** option lets you override Google Charts' can be either **`'line'`** or **`'point'`**. **Type:** string **Default:** 'point'
annotations.textStyle	For charts that support annotations (https://developers.google.com/chart/interactive/docs/roles), object controls the appearance of the text of the annotation: `var options = {` `  annotations: {`

```
    textStyle: {
      fontName: 'Times-Roman',
      fontSize: 18,
      bold: true,
      italic: true,
      // The color of the text.
      color: '#871b47',
      // The color of the text outline.
      auraColor: '#d799ae',
      // The transparency of the text.
      opacity: 0.8
    }
  }
};
```

This option is currently supported for area, bar, column, combo, supported by the Annotation Chart (https://developers.google.com/chart/interactive/docs/gallery,

Type: object
Default: null

axisTitlesPosition	Where to place the axis titles, compared to the chart area. Supp • in - Draw the axis titles inside the chart area. • out - Draw the axis titles outside the chart area. • none - Omit the axis titles. **Type:** string **Default:** 'out'
backgroundColor	The background color for the main area of the chart. Can be eitl for example: `'red'` or `'#00cc00'`, or an object with the follow **Type:** string or object **Default:** 'white'
backgroundColor.stroke	The color of the chart border, as an HTML color string. **Type:** string

	Default: '#666'
backgroundColor.strokeWidth	The border width, in pixels. **Type:** number **Default:** 0
backgroundColor.fill	The chart fill color, as an HTML color string. **Type:** string **Default:** 'white'
chartArea	An object with members to configure the placement and size of itself is drawn, excluding axis and legends). Two formats are su followed by %. A simple number is a value in pixels; a number fo Example: `chartArea:{left:20,top:0,width:'50%',hei` **Type:** object **Default:** null
chartArea.backgroundColor	Chart area background color. When a string is used, it can be eit an English color name. When an object is used, the following pr • `stroke`: the color, provided as a hex string or English color r • `strokeWidth`: if provided, draws a border around the chart the color of `stroke`). **Type:** string or object **Default:** 'white'
chartArea.left	How far to draw the chart from the left border. **Type:** number or string **Default:** auto
chartArea.top	How far to draw the chart from the top border. **Type:** number or string **Default:** auto
chartArea.width	Chart area width. **Type:** number or string **Default:** auto
chartArea.height	Chart area height. **Type:** number or string **Default:** auto
colors	The colors to use for the chart elements. An array of strings, wh color string, for example: `colors:['red','#004411']`.

	Type: Array of strings **Default:** default colors
crosshair	An object containing the crosshair (https://google-developers.appspot.com/chart/interactive/doc chart. **Type:** object **Default:** null
crosshair.color	The crosshair color, expressed as either a color name (e.g., "blu **Type:** string **Type:** default
crosshair.focused	An object containing the crosshair properties upon focus. Example: `crosshair: { focused: { color: '#3bc', c` **Type:** object **Default:** default
crosshair.opacity	The crosshair opacity, with `0.0` being fully transparent and `1.0` **Type:** number **Default:** 1.0
crosshair.orientation	The crosshair orientation, which can be 'vertical' for vertical hair hairs only, or 'both' for traditional crosshairs. **Type:** string **Default:** 'both'
crosshair.selected	An object containing the crosshair properties upon selection. Example: `crosshair: { selected: { color: '#3bc',` **Type:** object **Default:** default
crosshair.trigger	When to display crosshairs: on `'focus'`, `'selection'`, or `'b` **Type:** string **Default:** 'both'
curveType	Controls the curve of the lines when the line width is not zero. C • 'none' - Straight lines without curve. • 'function' - The angles of the line will be smoothed. **Type:**string **Default:** 'none'
dataOpacity	The transparency of data points, with 1.0 being completely opac

	scatter, histogram, bar, and column charts, this refers to the visi and rectangles in the others. In charts where *selecting data* crea area charts, this refers to the circles that appear upon hover or s exhibits both behaviors, and this option has no effect on other c a trendline, see trendline opacity (https://developers.google.com/chart/interactive/docs/gallery/ **Type:** number **Default:** 1.0
enableInteractivity	Whether the chart throws user-based events or reacts to user in not throw 'select' or other interaction-based events (but *will* thro not display hovertext or otherwise change depending on user in **Type:** boolean **Default:** true
explorer	The **`explorer`** option allows users to pan and zoom Google ch the default explorer behavior, enabling users to pan horizontally to zoom in and out by scrolling. This feature is **experimental** and may change in future releases. **Note:** The explorer only works with continuous axes (such as nu **Type:** object **Default:** null
explorer.actions	The Google Charts explorer supports three actions: • **`dragToPan`**: Drag to pan around the chart horizontally and v horizontal axis, use **`explorer: { axis: 'horizontal'`** axis. • **`dragToZoom`**: The explorer's default behavior is to zoom in a **`explorer: { actions: ['dragToZoom', 'rightCl`** dragging across a rectangular area zooms into that area. We **`rightClickToReset`** whenever **`dragToZoom`** is used. See **`explorer.maxZoomOut`**, and **`explorer.zoomDelta`** for z • **`rightClickToReset`**: Right clicking on the chart returns it level. **Type:** Array of strings **Default:** ['dragToPan', 'rightClickToReset']
explorer.axis	By default, users can pan both horizontally and vertically when t you want to users to only pan horizontally, use **`explorer: { a`** Similarly, **`explorer: { axis: 'vertical' }`** enables verti **Type:** string

	Default: both horizontal and vertical panning
explorer.keepInBounds	By default, users can pan all around, regardless of where the da pan beyond the original chart, use `explorer: { keepInBou` **Type:** boolean **Default:** false
explorer.maxZoomIn	The maximum that the explorer can zoom in. By default, users v that they'll see only 25% of the original view. Setting `explorer` let users zoom in only far enough to see half of the original view **Type:** number **Default:** 0.25
explorer.maxZoomOut	The maximum that the explorer can zoom out. By default, users enough that the chart will take up only 1/4 of the available spac `maxZoomOut: 8 }` would let users zoom out far enough that t of the available space. **Type:** number **Default:** 4
explorer.zoomDelta	When users zoom in or out, `explorer.zoomDelta` determines smaller the number, the smoother and slower the zoom. **Type:** number **Default:** 1.5
focusTarget	The type of the entity that receives focus on mouse hover. Also by mouse click, and which data table element is associated with following: • 'datum' - Focus on a single data point. Correlates to a cell in • 'category' - Focus on a grouping of all data points along the r the data table. In focusTarget 'category' the tooltip displays all the category val comparing values of different series. **Type:** string **Default:** 'datum'
fontSize	The default font size, in pixels, of all text in the chart. You can o specific chart elements. **Type:** number **Default:** automatic
fontName	The default font face for all text in the chart. You can override th chart elements.

	Type: string **Default:** 'Arial'
forceIFrame	Draws the chart inside an inline frame. (Note that on IE8, this op drawn in i-frames.) **Type:** boolean **Default:** false
hAxis	An object with members to configure various horizontal axis ele this object, you can use object literal notation, as shown here: `{` `  title: 'Hello',` `  titleTextStyle: {` `  color: '#FF0000'` `  }` `}` **Type:** object **Default:** null
hAxis.baseline	The baseline for the horizontal axis. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** automatic
hAxis.baselineColor	The color of the baseline for the horizontal axis. Can be any HTI `'red'` or `'#00cc00'`. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** 'black'
hAxis.direction	The direction in which the values along the horizontal axis grow of the values. **Type:** 1 or -1 **Default:** 1

hAxis.format	A format string for numeric or date axis labels. For number axis labels, this is a subset of the decimal formattin (http://icu-project.org/apiref/icu4c/classDecimalFormat.html#_ `{format:'#,###%'}` will display values "1,000%", "750%", and You can also supply any of the following: • `{format: 'none'}`: displays numbers with no formatting • `{format: 'decimal'}`: displays numbers with thousands • `{format: 'scientific'}`: displays numbers in scientific • `{format: 'currency'}`: displays numbers in the local cu • `{format: 'percent'}`: displays numbers as percentages • `{format: 'short'}`: displays abbreviated numbers (e.g., • `{format: 'long'}`: displays numbers as full words (e.g., For date axis labels, this is a subset of the date formatting ICU p (http://icu-project.org/apiref/icu4c/classSimpleDateFormat.htr `{format:'MMM d, y'}` will display the value "Jul 1, 2011" for The actual formatting applied to the label is derived from the loc with. For more details, see loading charts with a specific locale (https://google-developers.appspot.com/chart/interactive/docs/library_loading . This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** string **Default:** auto
hAxis.gridlines	An object with members to configure the gridlines on the horizo of this object, you can use object literal notation, as shown here `{color: '#333', count: 4}` This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** object **Default:** null
hAxis.gridlines.color	The color of the horizontal gridlines inside the chart area. Speci

	Type: string **Default:** '#CCC'
hAxis.gridlines.count	The number of horizontal gridlines inside the chart area. Minimu automatically compute the number of gridlines. **Type:** number **Default:** 5
hAxis.gridlines.units	Overrides the default format for various aspects of date/datetin used with chart computed gridlines. Allows formatting for years seconds, and milliseconds. General format is: `gridlines: {` `  units: {` `  years: {format: [/*format strings here*/]},` `  months: {format: [/*format strings here*/]}` `  days: {format: [/*format strings here*/]}` `  hours: {format: [/*format strings here*/]}` `  minutes: {format: [/*format strings here*/]` `  seconds: {format: [/*format strings here*/]` `  milliseconds: {format: [/*format strings he` `  }` `}` Additional information can be found in Dates and Times (https://google-developers.appspot.com/chart/interactive/docs/datesandtimes . **Type:** object **Default:** null
hAxis.minorGridlines	An object with members to configure the minor gridlines on the hAxis.gridlines option. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** object **Default:** null
hAxis.minorGridlines.color	The color of the horizontal minor gridlines inside the chart area. string.

	Type: string **Default:** A blend of the gridline and background colors
hAxis.minorGridlines.count	The number of horizontal minor gridlines between two regular g **Type:** number **Default:** 0
hAxis.minorGridlines.units	Overrides the default format for various aspects of date/datetin used with chart computed minorGridlines. Allows formatting fo minutes, seconds, and milliseconds. General format is: `gridlines: {` `  units: {` `  years: {format: [/*format strings here*/]},` `  months: {format: [/*format strings here*/]}` `  days: {format: [/*format strings here*/]}` `  hours: {format: [/*format strings here*/]}` `  minutes: {format: [/*format strings here*/]` `  seconds: {format: [/*format strings here*/]` `  milliseconds: {format: [/*format strings he` `  }` `}` Additional information can be found in Dates and Times (https://google-developers.appspot.com/chart/interactive/docs/datesandtimes . **Type:** object **Default:** null
hAxis.logScale	**`hAxis`** property that makes the horizontal axis a logarithmic sc positive). Set to true for yes. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** boolean **Default:** false
hAxis.scaleType	**`hAxis`** property that makes the horizontal axis a logarithmic sc • null - No logarithmic scaling is performed.

	• 'log' - Logarithmic scaling. Negative and zero values are not as setting **`hAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero value of a negative number is the negative of the log of the a are plotted on a linear scale. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** string **Default: null**
hAxis.textPosition	Position of the horizontal axis text, relative to the chart area. Su **Type:** string **Default:** 'out'
hAxis.textStyle	An object that specifies the horizontal axis text style. The objec `{ color: <string>,` `fontName: <string>,` `fontSize: <number>,` `bold: <boolean>,` `italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default: `{color: 'black', fontName: <global-font-r <global-font-size>}`**
hAxis.ticks	Replaces the automatically generated X-axis ticks with the spec array should be either a valid tick value (such as a number, date object. If it's an object, it should have a **v** property for the tick va containing the literal string to be displayed as the label. Examples: • **`hAxis: { ticks: [5,10,15,20] }`** • **`hAxis: { ticks: [{v:32, f:'thirty two'}, {v:`** • **`hAxis: { ticks: [new Date(2014,3,15), new Da`** • **`hAxis: { ticks: [16, {v:32, f:'thirty two'},`** **`128] }`**

	This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** Array of elements **Default:** auto
hAxis.title	**`hAxis`** property that specifies the title of the horizontal axis. **Type:** string **Default:** null
hAxis.titleTextStyle	An object that specifies the horizontal axis title text style. The o `{ color: <string>,` `fontName: <string>,` `fontSize: <number>,` `bold: <boolean>,` `italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-r <global-font-size>}`**
hAxis.allowContainerBoundaryTextCufoff	If false, will hide outermost labels rather than allow them to be c true, will allow label cropping. This option is only supported for a `discrete` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** boolean **Default:** false
hAxis.slantedText	If true, draw the horizontal axis text at an angle, to help fit more horizontal axis text upright. Default behavior is to slant text if it upright. Notice that this option is available only when the **`hAxis`** (which is the default). This option is only supported for a `discrete` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis.

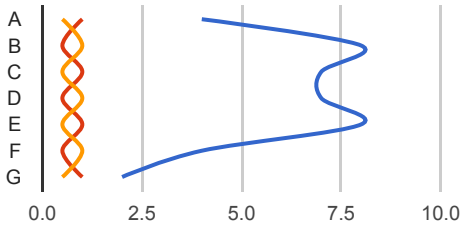
	Type: boolean **Default:** automatic
hAxis.slantedTextAngle	The angle of the horizontal axis text, if it's drawn slanted. Ignore `false`, or is in auto mode, and the chart decided to draw the tex This option is only supported for a discrete (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number, 1—90 **Default:** 30
hAxis.maxAlternation	Maximum number of levels of horizontal axis text. If axis text la server might shift neighboring labels up or down in order to fit la specifies the most number of levels to use; the server can use f without overlapping. This option is only supported for a discrete (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** 2
hAxis.maxTextLines	Maximum number of lines allowed for the text labels. Labels ca too long, and the number of lines is, by default, limited by the he This option is only supported for a discrete (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** auto
hAxis.minTextSpacing	Minimum horizontal spacing, in pixels, allowed between two adj are spaced too densely, or they are too long, the spacing can dr this case one of the label-unclutter measures will be applied (e. dropping some of them). This option is only supported for a discrete (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** The value of `hAxis.textStyle.fontSize`

hAxis.showTextEvery	How many horizontal axis labels to show, where 1 means show other label, and so on. Default is to try to show as many labels a This option is only supported for a `discrete` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** automatic
hAxis.maxValue	Moves the max value of the horizontal axis to the specified valu charts. Ignored if this is set to a value smaller than the maximu **`hAxis.viewWindow.max`** overrides this property. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** automatic
hAxis.minValue	Moves the min value of the horizontal axis to the specified value charts. Ignored if this is set to a value greater than the minimum **`hAxis.viewWindow.min`** overrides this property. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** number **Default:** automatic
hAxis.viewWindowMode	Specifies how to scale the horizontal axis to render the values w following string values are supported: • 'pretty' - Scale the horizontal values so that the maximum an rendered a bit inside the left and right of the chart area. This **`haxis.viewWindow.min`** and **`haxis.viewWindow.max`** t • 'maximized' - Scale the horizontal values so that the maximu touch the left and right of the chart area. This will cause **hax** **`haxis.viewWindow.max`** to be ignored. • 'explicit' - A deprecated option for specifying the left and righ (Deprecated because it's redundant with **`haxis.viewWind`** **`haxis.viewWindow.max`**.) Data values outside these value specify an **`hAxis.viewWindow`** object describing the maxi show.

	This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** string **Default:** Equivalent to 'pretty', but **`haxis.viewWindow.min`** an take precedence if used.
hAxis.viewWindow	Specifies the cropping range of the horizontal axis. **Type:** object **Default:** null
hAxis.viewWindow.max	• For a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizin axis: The maximum horizontal data value to render. • For a `discrete` (https://google-developers.appspot.com/chart/interactive/docs/customizin axis: The zero-based row index where the cropping window ends. higher will be cropped out. In conjunction with **`vAxis.view`** half-opened range [min, max) that denotes the element indic every index such that **`min <= index < max`** will be displa Ignored when **`hAxis.viewWindowMode`** is 'pretty' or 'maximize **Type:** number **Default:** auto
hAxis.viewWindow.min	• For a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizin axis: The minimum horizontal data value to render. • For a `discrete` (https://google-developers.appspot.com/chart/interactive/docs/customizin axis: The zero-based row index where the cropping window begin than this will be cropped out. In conjunction with **`vAxis.vi`** a half-opened range [min, max) that denotes the element ind every index such that **`min <= index < max`** will be displa Ignored when **`hAxis.viewWindowMode`** is 'pretty' or 'maximize

	Type: number **Default:** auto
height	Height of the chart, in pixels. **Type:** number **Default:** height of the containing element
interpolateNulls	Whether to guess the value of missing points. If true, it will gues based on neighboring points. If false, it will leave a break in the This **is not** supported by Area (https://google-developers.appspot.com/chart/interactive/doc the `isStacked: true/'percent'/'relative'/'absolu` **Type:** boolean **Default:** false
legend	An object with members to configure various aspects of the leg object, you can use object literal notation, as shown here: `{position: 'top', textStyle: {color: 'blue', fo` **Type:** object **Default:** null
legend.alignment	Alignment of the legend. Can be one of the following: • 'start' - Aligned to the start of the area allocated for the leger • 'center' - Centered in the area allocated for the legend. • 'end' - Aligned to the end of the area allocated for the legend. Start, center, and end are relative to the style -- vertical or horizo example, in a 'right' legend, 'start' and 'end' are at the top and bo legend, 'start' and 'end' would be at the left and right of the area, The default value depends on the legend's position. For 'bottom other legends default to 'start'. **Type:** string **Default:** automatic
legend.maxLines	Maximum number of lines in the legend. Set this to a number gr your legend. Note: The exact logic used to determine the actual in flux. This option currently works only when legend.position is 'top'. **Type:** number

	Default: 1
legend.position	Position of the legend. Can be one of the following: • 'bottom' - Below the chart. • 'left' - To the left of the chart, provided the left axis has no se want the legend on the left, use the option **`targetAxisInd`** • 'in' - Inside the chart, by the top left corner. • 'none' - No legend is displayed. • 'right' - To the right of the chart. Incompatible with the **`vAxes`** • 'top' - Above the chart. **Type:** string **Default:** 'right'
legend.textStyle	An object that specifies the legend text style. The object has thi `{ color: <string>,` `fontName: <string>,` `fontSize: <number>,` `bold: <boolean>,` `italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-n <global-font-size>}`**
lineDashStyle	The on-and-off pattern for dashed lines. For instance, **`[4, 4]`** w followed by 4-length gaps, and **`[5, 1, 3]`** will repeat a 5-lengt dash, a 5-length gap, a 1-length dash, and a 3-length gap. See Da (https://developers.google.com/chart/interactive/docs/lines#c **Type:** Array of numbers **Default:** null
lineWidth	Data line width in pixels. Use zero to hide all lines and show only values for individual series using the **`series`** property. **Type:** number **Default:** 2
orientation	The orientation of the chart. When set to **`'vertical'`**, rotates instance) a column chart becomes a bar chart, and an area cha

	 **Type:** string **Default:** 'horizontal'
pointShape	The shape of individual data elements: 'circle', 'triangle', 'square', See the points documentation (https://developers.google.com/ for examples. **Type:** string **Default:** 'circle'
pointSize	Diameter of displayed points in pixels. Use zero to hide all point individual series using the **`series`** property. If you're using a tre (https://google-developers.appspot.com/chart/interactive/doc **`pointSize`** option will affect the width of the trendline unless y **`trendlines.n.pointsize`** option. **Type:** number **Default:** 0
pointsVisible	Determines whether points will be displayed. Set to **`false`** to hi values for individual series using the **`series`** property. If you're (https://google-developers.appspot.com/chart/interactive/doc **`pointsVisible`** option will affect the visibility of the points on override it with the **`trendlines.n.pointsVisible`** option. This can also be overridden using the style role (https://google-developers.appspot.com/chart/interactive/doc "**`point {visible: true}`**". **Type:** boolean **Default:** true
reverseCategories	If set to true, will draw series from right to left. The default is to This option is only supported for a **`discrete`** (https://google-developers.appspot.com/chart/interactive/docs/customizing_a **`major`** (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** boolean

	Default: false
selectionMode	When **`selectionMode`** is **`'multiple'`**, users may select mult **Type:** string **Default:** 'single'
series	An array of objects, each describing the format of the correspor default values for a series, specify an empty object {}. If a series global value will be used. Each object supports the following pr • **`annotations`** - An object to be applied to annotations for th control, for instance, the **`textStyle`** for the series: `series: { 0: { annotations: { textStyle: {fontSize: 12, color: 'red' } } } }` See the various **`annotations`** options for a more complete • **`color`** - The color to use for this series. Specify a valid HTM • **`curveType`** - Overrides the global **`curveType`** value for this • **`labelInLegend`** - The description of the series to appear in • **`lineDashStyle`** - Overrides the global **`lineDashStyle`** va • **`lineWidth`** - Overrides the global **`lineWidth`** value for this • **`pointShape`** - Overrides the global **`pointShape`** value for t • **`pointSize`** - Overrides the global **`pointSize`** value for this • **`pointsVisible`** - Overrides the global **`pointsVisible`** va • **`targetAxisIndex`** - Which axis to assign this series to, wh the opposite axis. Default value is 0; set to 1 to define a char rendered against different axes. At least one series much be You can define a different scale for different axes. • **`visibleInLegend`** - A boolean value, where true means tha legend entry, and false means that it should not. Default is tr You can specify either an array of objects, each of which applies or you can specify an object where each child has a numeric ke applies to. For example, the following two declarations are iden as black and absent from the legend, and the fourth as red and

	`` series: [{color: 'black', visibleInLegend: false}, {}, {color: 'red', visibleInLegend: false}] series: { 0:{color: 'black', visibleInLegend: false}, 3:{color: 'red', visibleInLegend: false} } ``` **Type:** Array of objects, or object with nested objects **Default:** {}
theme	A theme is a set of predefined option values that work together behavior or visual effect. Currently only one theme is available: • 'maximized' - Maximizes the area of the chart, and draws the inside the chart area. Sets the following options: ``` chartArea: {width: '100%', height: '100%'}, legend: {position: 'in'}, titlePosition: 'in', axisTitlesPosition: 'in hAxis: {textPosition: 'in'}, vAxis: {textPos ``` **Type:** string **Default:** null
title	Text to display above the chart. **Type:** string **Default:** no title
titlePosition	Where to place the chart title, compared to the chart area. Supp • in - Draw the title inside the chart area. • out - Draw the title outside the chart area. • none - Omit the title. **Type:** string **Default:** 'out'
titleTextStyle	An object that specifies the title text style. The object has this fo ``` { color: <string>, fontName: <string>, fontSize: <number>, bold: <boolean>, ```

	`italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-r <global-font-size>}`**
tooltip	An object with members to configure various tooltip elements. T object, you can use object literal notation, as shown here: `{textStyle: {color: '#FF0000'}, showColorCode:` **Type:** object **Default:** null
tooltip.ignoreBounds	If set to **`true`**, allows the drawing of tooltips to flow outside of t sides. **Note:** This only applies to HTML tooltips. If this is enabled with outside of the chart bounds will be cropped. See Customizing T (https://google-developers.appspot.com/chart/interactive/doc for more details. **Type:** boolean **Default:** false
tooltip.isHtml	If set to true, use HTML-rendered (rather than SVG-rendered) too Content (https://google-developers.appspot.com/chart/interactive/doc for more details. ★ **Note:** customization of the HTML tooltip content via the tooltip (https://google-developers.appspot.com/chart/interactive/doc supported by the Bubble Chart (https://google-developers.appspot.com/chart/interactive/doc visualization. **Type:** boolean **Default:** false
tooltip.showColorCode	If true, show colored squares next to the series information in th when **`focusTarget`** is set to 'category', otherwise the default is **Type:** boolean **Default:** automatic

tooltip.textStyle	An object that specifies the tooltip text style. The object has this `{ color: <string>,` `  fontName: <string>,` `  fontSize: <number>,` `  bold: <boolean>,` `  italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-n <global-font-size>}`**
tooltip.trigger	The user interaction that causes the tooltip to be displayed: • 'focus' - The tooltip will be displayed when the user hovers ov • 'none' - The tooltip will not be displayed. • 'selection' - The tooltip will be displayed when the user select **Type:** string **Default:** 'focus'
trendlines	Displays trendlines (https://developers.google.com/chart/inter on the charts that support them. By default, linear trendlines are customized with the **`trendlines.`*`n`*`.type`** option. Trendlines are specified on a per-series basis, so most of the tir `var options = {` `  trendlines: {` `  0: {` `  type: 'linear',` `  color: 'green',` `  lineWidth: 3,` `  opacity: 0.3,` `  showR2: true,` `  visibleInLegend: true` `  }` `  }` `}` **Type:** object **Default:** null
trendlines.n.color	The color of the trendline

	(https://developers.google.com/chart/interactive/docs/gallery/ an English color name or a hex string. **Type:** string **Default:** default series color
trendlines.n.degree	For trendlines (https://developers.google.com/chart/interactive **`type: 'polynomial'`**, the degree of the polynomial (**2** for qu (The default degree may change from 3 to 2 in an upcoming rele **Type:** number **Default:** 3
trendlines.n.labelInLegend	If set, the trendline (https://developers.google.com/chart/intera will appear in the legend as this string. **Type:** string **Default:** null
trendlines.n.lineWidth	The line width of the trendline (https://developers.google.com/chart/interactive/docs/gallery/ **Type:** number **Default:** 2
trendlines.n.opacity	The transparency of the trendline (https://developers.google.com/chart/interactive/docs/gallery/ (transparent) to 1.0 (opaque). **Type:** number **Default:** 1.0
trendlines.n.pointSize	Trendlines (https://developers.google.com/chart/interactive/do constucted by stamping a bunch of dots on the chart; this rarely customize the size of the dots. The trendline's **`lineWidth`** opti However, you'll need this option if you're using the global **`point`** different point size for your trendlines. **Type:** number **Default:** 1
trendlines.n.pointsVisible	Trendlines (https://developers.google.com/chart/interactive/do constucted by stamping a bunch of dots on the chart. The trend determines whether the points for a particular trendline are visib **Type:** boolean **Default:** true
trendlines.n.showR2	Whether to show the coefficient of determination (https://developers.google.com/chart/interactive/docs/gallery/ trendline tooltip.

	Type: boolean **Default:** false
trendlines.n.type	Whether the trendlines (https://developers.google.com/chart/interactive/docs/gallery/ default), `'exponential'`, or `'polynomial'`. **Type:** string **Default:** linear
trendlines.n.visibleInLegend	Whether the trendline (https://developers.google.com/chart/interactive/docs/gallery/ the legend. (It will appear in the trendline tooltip.) **Type:** boolean **Default:** false
vAxes	Specifies properties for individual vertical axes, if the chart has object is a **`vAxis`** object, and can contain all the properties sup values override any global settings for the same property. To specify a chart with multiple vertical axes, first define a new **`series.targetAxisIndex`**, then configure the axis using **`vA`** assigns series 2 to the right axis and specifies a custom title an `{ series: { 2: { targetAxisIndex:1 } }, vAxes: { 1: { title:'Losses', textStyle: {color: 'red'} } } }` This property can be either an object or an array: the object is a numeric label that specifies the axis that it defines--this is the fo an array of objects, one per axis. For example, the following arra the **`vAxis`** object shown above: `vAxes: [ {}, // Nothing specified for axis 0 { title:'Losses',`

	<pre> textStyle: {color: 'red'} // Axis 1 }]</pre> **Type:** Array of object, or object with child objects **Default:** null
vAxis	An object with members to configure various vertical axis eleme object, you can use object literal notation, as shown here: <pre>{title: 'Hello', titleTextStyle: {color: '#FF00</pre> **Type:** object **Default:** null
vAxis.baseline	**`vAxis`** property that specifies the baseline for the vertical axis. highest grid line or smaller than the lowest grid line, it will be rou **Type:** number **Default:** automatic
vAxis.baselineColor	Specifies the color of the baseline for the vertical axis. Can be a example: **`'red'`** or **`'#00cc00'`**. **Type:** number **Default:** 'black'
vAxis.direction	The direction in which the values along the vertical axis grow. Sp the values. **Type:** 1 or -1 **Default:** 1
vAxis.format	A format string for numeric axis labels. This is a subset of the IC (http://icu-project.org/apiref/icu4c/classDecimalFormat.html#_ **`{format:'#,###%'}`** will display values "1,000%", "750%", and You can also supply any of the following: • **`{format: 'none'}`**: displays numbers with no formatting • **`{format: 'decimal'}`**: displays numbers with thousands • **`{format: 'scientific'}`**: displays numbers in scientific • **`{format: 'currency'}`**: displays numbers in the local cu • **`{format: 'percent'}`**: displays numbers as percentages • **`{format: 'short'}`**: displays abbreviated numbers (e.g., • **`{format: 'long'}`**: displays numbers as full words (e.g.,

	The actual formatting applied to the label is derived from the loc with. For more details, see loading charts with a specific locale (https://google-developers.appspot.com/chart/interactive/docs/library_loading . **Type:** string **Default:** auto
vAxis.gridlines	An object with members to configure the gridlines on the vertica this object, you can use object literal notation, as shown here: `{color: '#333', count: 4}` **Type:** object **Default:** null
vAxis.gridlines.color	The color of the vertical gridlines inside the chart area. Specify a **Type:** string **Default:** '#CCC'
vAxis.gridlines.count	The number of vertical gridlines inside the chart area. Minimum automatically compute the number of gridlines. **Type:** number **Default:** 5
vAxis.gridlines.units	Overrides the default format for various aspects of date/datetin used with chart computed gridlines. Allows formatting for years seconds, and milliseconds. General format is: `gridlines: {` `  units: {` `  years: {format: [/*format strings here*/]},` `  months: {format: [/*format strings here*/]}` `  days: {format: [/*format strings here*/]}` `  hours: {format: [/*format strings here*/]}` `  minutes: {format: [/*format strings here*/]` `  seconds: {format: [/*format strings here*/]` `  milliseconds: {format: [/*format strings he` `  }` `}`

	Additional information can be found in Dates and Times (https://google-developers.appspot.com/chart/interactive/docs/datesandtimes . **Type:** object **Default:** null
vAxis.minorGridlines	An object with members to configure the minor gridlines on the vAxis.gridlines option. **Type:** object **Default:** null
vAxis.minorGridlines.color	The color of the vertical minor gridlines inside the chart area. Sp **Type:** string **Default:** A blend of the gridline and background colors
vAxis.minorGridlines.count	The number of vertical minor gridlines between two regular grid **Type:** number **Default:** 0
vAxis.minorGridlines.units	Overrides the default format for various aspects of date/datetin used with chart computed minorGridlines. Allows formatting fo minutes, seconds, and milliseconds. General format is: `gridlines: {` `  units: {` `  years: {format: [/*format strings here*/]},` `  months: {format: [/*format strings here*/]}` `  days: {format: [/*format strings here*/]}` `  hours: {format: [/*format strings here*/]}` `  minutes: {format: [/*format strings here*/]` `  seconds: {format: [/*format strings here*/]` `  milliseconds: {format: [/*format strings he` `  }` `}` Additional information can be found in Dates and Times (https://google-developers.appspot.com/chart/interactive/docs/datesandtimes . **Type:** object **Default:** null

vAxis.logScale	If true, makes the vertical axis a logarithmic scale. Note: All valu **Type:** boolean **Default:** false
vAxis.scaleType	**`vAxis`** property that makes the vertical axis a logarithmic scale • null - No logarithmic scaling is performed. • 'log' - Logarithmic scaling. Negative and zero values are not as setting **`vAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero value of a negative number is the negative of the log of the a are plotted on a linear scale. This option is only supported for a `continuous` (https://google-developers.appspot.com/chart/interactive/docs/customizing_a axis. **Type:** string **Default:** null
vAxis.textPosition	Position of the vertical axis text, relative to the chart area. Supp **Type:** string **Default:** 'out'
vAxis.textStyle	An object that specifies the vertical axis text style. The object ha `{ color: <string>,` `fontName: <string>,` `fontSize: <number>,` `bold: <boolean>,` `italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-r <global-font-size>}`**
vAxis.ticks	Replaces the automatically generated Y-axis ticks with the spec array should be either a valid tick value (such as a number, date object. If it's an object, it should have a **v** property for the tick va containing the literal string to be displayed as the label. Examples:

	• `vAxis: { ticks: [5,10,15,20] }` • `vAxis: { ticks: [{v:32, f:'thirty two'}, {v:` • `vAxis: { ticks: [new Date(2014,3,15), new Da` • `vAxis: { ticks: [16, {v:32, f:'thirty two'},` `128] }` **Type:** Array of elements **Default:** auto
vAxis.title	**`vAxis`** property that specifies a title for the vertical axis. **Type:** string **Default:** no title
vAxis.titleTextStyle	An object that specifies the vertical axis title text style. The obje `{ color: <string>,` `fontName: <string>,` `fontSize: <number>,` `bold: <boolean>,` `italic: <boolean> }` The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontName`** and **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-r <global-font-size>}`**
vAxis.maxValue	Moves the max value of the vertical axis to the specified value; t charts. Ignored if this is set to a value smaller than the maximur **`vAxis.viewWindow.max`** overrides this property. **Type:** number **Default:** automatic
vAxis.minValue	Moves the min value of the vertical axis to the specified value; t charts. Ignored if this is set to a value greater than the minimum **`vAxis.viewWindow.min`** overrides this property. **Type:** number **Default:** null
vAxis.viewWindowMode	Specifies how to scale the vertical axis to render the values with string values are supported: • 'pretty' - Scale the vertical values so that the maximum and r rendered a bit inside the top and bottom of the chart area. Th

	vaxis.viewWindow.min and **vaxis.viewWindow.max** t • 'maximized' - Scale the vertical values so that the maximum the top and bottom of the chart area. This will cause **vaxis vaxis.viewWindow.max** to be ignored. • 'explicit' - A deprecated option for specifying the top and bot area. (Deprecated because it's redundant with **vaxis.view vaxis.viewWindow.max**. Data values outside these value specify a **vAxis.viewWindow** object describing the maxim show. **Type:** string **Default:** Equivalent to 'pretty', but **vaxis.viewWindow.min** an take precedence if used.
vAxis.viewWindow	Specifies the cropping range of the vertical axis. **Type:** object **Default:** null
vAxis.viewWindow.max	The maximum vertical data value to render. Ignored when **vAxis.viewWindowMode** is 'pretty' or 'maximize **Type:** number **Default:** auto
vAxis.viewWindow.min	The minimum horizontal data value to render. Ignored when **vAxis.viewWindowMode** is 'pretty' or 'maximize **Type:** number **Default:** auto
width	Width of the chart, in pixels. **Type:** number **Default:** width of the containing element

Methods

Method	
`draw(data, options)`	Draws the chart. The chart accepts further method calls only after the ready (#Events)event is fired. Extended description (https://google-developers.appspot.com/chart/interactive/docs/reference#visdraw) .

	Return Type: none
`getAction(actionID)`	Returns the tooltip action object with the requested `actionID`. **Return Type:** object
`getBoundingBox(id)`	Returns an object containing the left, top, width, and height of chart ele `id`. The format for `id` isn't yet documented (they're the return values o event handlers (https://developers.google.com/chart/interactive/docs/events)), but h are some examples: `var cli = chart.getChartLayoutInterface();` **Height of the chart area** `cli.getBoundingBox('chartarea').height` **Width of the third bar in the first series of a bar or column cl** `cli.getBoundingBox('bar#0#2').width` **Bounding box of the fifth wedge of a pie chart** `cli.getBoundingBox('slice#4')` **Bounding box of the chart data of a vertical (e.g., column) chart:** `cli.getBoundingBox('vAxis#0#gridline')` **Bounding box of the chart data of a horizontal (e.g., bar) cha** `cli.getBoundingBox('hAxis#0#gridline')` Values are relative to the container of the chart. Call this *after* the chart drawn. **Return Type:** object
`getChartAreaBoundingBox()`	Returns an object containing the left, top, width, and height of the chart content (i.e., excluding labels and legend): `var cli = chart.getChartLayoutInterface();` `cli.getChartAreaBoundingBox().left`

	`cli.getChartAreaBoundingBox().top` `cli.getChartAreaBoundingBox().height` `cli.getChartAreaBoundingBox().width` Values are relative to the container of the chart. Call this *after* the chart drawn. **Return Type:** object
`getChartLayoutInterface()`	Returns an object containing information about the onscreen placeme the chart and its elements. The following methods can be called on the returned object: • `getBoundingBox` • `getChartAreaBoundingBox` • `getHAxisValue` • `getVAxisValue` • `getXLocation` • `getYLocation` Call this *after* the chart is drawn. **Return Type:** object
`getHAxisValue(position, optional_axis_index)`	Returns the logical horizontal value at `position`, which is an offset fr the chart container's left edge. Can be negative. Example: `chart.getChartLayoutInterface().getHAxisValue(400)`. Call this *after* the chart is drawn. **Return Type:** number
`getImageURI()`	Returns the chart serialized as an image URI. Call this *after* the chart is drawn. See Printing PNG Charts (https://google-developers.appspot.com/chart/interactive/docs/printi **Return Type:** string
`getSelection()`	Returns an array of the selected chart entities. Selectable entities are p annotations, legend entries and categories. A point or annotation corresponds to a cell in the data table, a legend entry to a column (row

	index is null), and a category to a row (column index is null). For this ch only one entity can be selected at any given moment. Extended description (https://google-developers.appspot.com/chart/interactive/docs/reference#visgetsele . **Return Type:** Array of selection elements
`getVAxisValue(position, optional_axis_index)`	Returns the logical vertical value at **`position`**, which is an offset from chart container's top edge. Can be negative. Example: **`chart.getChartLayoutInterface().getVAxisValue(300)`**. Call this *after* the chart is drawn. **Return Type:** number
`getXLocation(position, optional_axis_index)`	Returns the screen x-coordinate of **`position`** relative to the chart's container. Example: **`chart.getChartLayoutInterface().getXLocation(400)`**. Call this *after* the chart is drawn. **Return Type:** number
`getYLocation(position, optional_axis_index)`	Returns the screen y-coordinate of **`position`** relative to the chart's container. Example: **`chart.getChartLayoutInterface().getYLocation(300)`**. Call this *after* the chart is drawn. **Return Type:** number
`removeAction(actionID)`	Removes the tooltip action with the requested **`actionID`** from the cha **Return Type: none**
`setAction(action)`	Sets a tooltip action to be executed when the user clicks on the action The **`setAction`** method takes an object as its action parameter. This object should specify 3 properties: **`id`**— the ID of the action being set, —the text that should appear in the tooltip for the action, and **`action`** function that should be run when a user clicks on the action text. Any and all tooltip actions should be set prior to calling the chart's **`dra`** method. Extended description

	(https://google-developers.appspot.com/chart/interactive/docs/reference#vissetactic . **Return Type: none**
`setSelection()`	Selects the specified chart entities. Cancels any previous selection. Selectable entities are points, annotations, legend entries and categori point or annotation corresponds to a cell in the data table, a legend ent a column (row index is null), and a category to a row (column index is r For this chart, only one entity can be selected at a time. `Extended description` (https://google-developers.appspot.com/chart/interactive/docs/reference#vissetsele . **Return Type:** none
`clearChart()`	Clears the chart, and releases all of its allocated resources. **Return Type:** none

Events

For more information on how to use these events, see Basic Interactivity (https://google-developers.appspot.com/chart/interactive/docs/basic_interactivity), Handling Events (https://google-developers.appspot.com/chart/interactive/docs/events), and Firing Events (https://google-developers.appspot.com/chart/interactive/docs/dev/events).

Name	
`animationfinish`	Fired when transition animation is complete. **Properties:** none
`click`	Fired when the user clicks inside the chart. Can be used to identify when the title, data elements, legend entries, axes, gridlines, or labels are clicked. **Properties:** targetID
`error`	Fired when an error occurs when attempting to render the chart. **Properties:** id, message
`onmouseover`	Fired when the user mouses over a visual entity. Passes back the row and column indices of the corresponding data table element.

	Properties: row, column
`onmouseout`	Fired when the user mouses away from a visual entity. Passes back the row and column indices of the corresponding data table element. **Properties:** row, column
`ready`	The chart is ready for external method calls. If you want to interact with the chart, and call methods after you draw it, you should set up a listener for this event *before* you call the `draw` method, and call them only after the event was fired. **Properties:** none
`select`	Fired when the user clicks a visual entity. To learn what has been selected, call `getSelection()` (#Methods). **Properties:** none

Data Policy

All code and data are processed and rendered in the browser. No data is sent to any server.

上次更新日期：二月 23, 2017