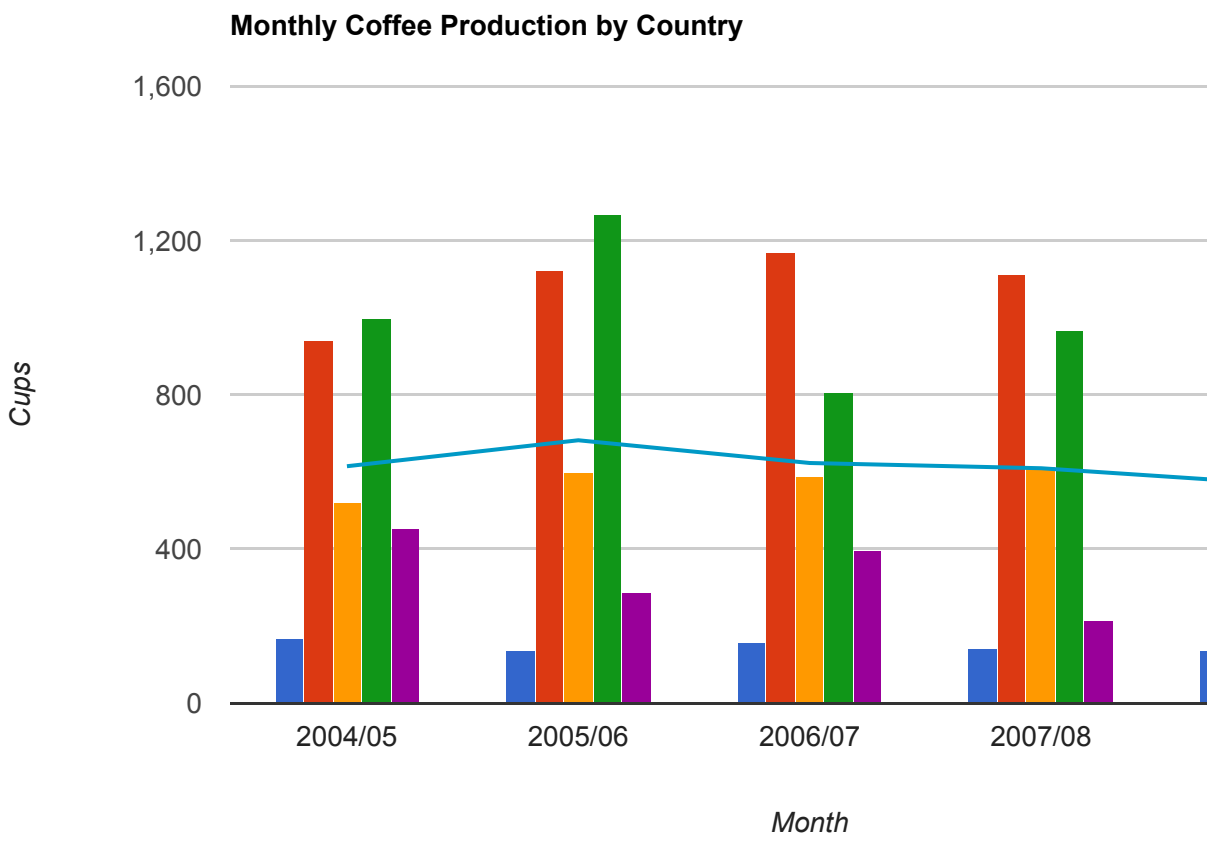
Visualization: Combo Chart

Overview

A chart that lets you render each series as a different marker type from the following list: line, area, bars, candlesticks, and stepped area.

To assign a default marker type for series, specify the `seriesType` property. Use the `series` property to specify properties of each series individually.

Example



CODE IT YOURSELF ON JSFIDDLE

```
<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(drawVisualization);

      function drawVisualization() {
        // Some raw data (not necessarily accurate)
        var data = google.visualization.arrayToDataTable([
         ['Month', 'Bolivia', 'Ecuador', 'Madagascar', 'Papua New Guinea', 'Rw
         ['2004/05',  165,      938,         522,             998,           4
         ['2005/06',  135,      1120,        599,             1268,          1
         ['2006/07',  157,      1167,        587,             807,           3
         ['2007/08',  139,      1110,        615,             968,           2
         ['2008/09',  136,      691,         629,             1026,          3
      ]);

    var options = {
      title : 'Monthly Coffee Production by Country',
      vAxis: {title: 'Cups'},
      hAxis: {title: 'Month'},
      seriesType: 'bars',
      series: {5: {type: 'line'}}
    };

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

Loading

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

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

The visualization's class name is **`google.visualization.ComboChart`**

```
var visualization = new google.visualization.ComboChart(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://developers.google.com/chart/i) • X-axis values (continuous (https://developers.google.com/chart/i)
Data Type:	• string (discrete (https://developers.google.com/chart/i) • number, date, datetime, or timeofday (continuous (https://developers.google.com/chart/i)
Role:	domain
Optional column roles (https://developers.google.com/chart/interactive/docs/roles) :	• annotation (https://developers.google.c • annotationText (https://developers.goo

Configuration Options

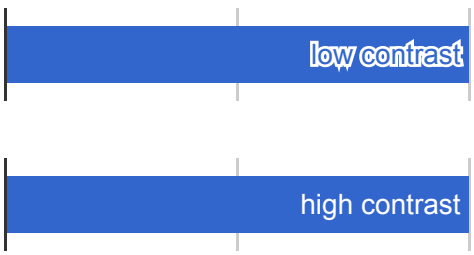
Name	

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`
	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://developers.google.com/chart/interactive/docs/animat **Type:** number **Default:** 0
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'

animation.startup	Determines if the chart will animate on the initial draw. If **`true`**, animate to its final state. **Type:** boolean **Default** false
annotations.boxStyle	For charts that support annotations (https://developers.google. **`annotations.boxStyle`** object controls the appearance of th

```
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://developers.google.com/chart/interactive/docs/gallery/ **Type:** object **Default:** null
annotations.datum	For charts that support annotations (https://developers.google. **`annotations.datum`** object lets you override Google Charts' d individual data elements (such as values displayed with each ba color with **`annotations.datum.stem.color`**, the stem lengt **`annotations.datum.stem.length`**, and the style with **`annc`** **Type:** object **Default:** color is "black"; length is 12; style is "point".
annotations.domain	For charts that support annotations (https://developers.google. **`annotations.domain`** object lets you override Google Charts' domain (the major axis of the chart, such as the X axis on a typi with **`annotations.domain.stem.color`**, the stem length wi **`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. **`annotations.highContrast`** boolean lets you override Goog color. By default, **`annotations.highContrast`** is true, which color with good contrast: light colors on dark backgrounds, and **`annotations.highContrast`** to false and don't specify your will use the default series color for the annotation:  **Type:** boolean **Default:** true
annotations.stem	For charts that support annotations (https://developers.google.

	`annotations.stem` object lets you override Google Charts' ch color with **`annotations.stem.color`** and the stem length wi that the stem length option has no effect on annotations with st annotations, the stem length is always the same as the text, and stem extends across the entire chart. **Type:** object **Default:** color is "black"; length is 5 for domain annotations and
annotations.style	For charts that support annotations (https://developers.google. **`annotations.style`** option lets you override Google Charts' d either **`'line'`** or **`'point'`**. **Type:** string **Default:** 'point'

annotations.textStyle	For charts that support annotations (https://developers.google.
	`annotations.textStyle` object controls the appearance of
	
	`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
areaOpacity	The default opacity of the colored area under an area chart seri is fully opaque. To specify opacity for an individual series, set th property. **Type:** number, 0.0–1.0 **Default:** 0.3
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 eith example: `'red'` or `'#00cc00'`, or an object with the following **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'
bar.groupWidth	The width of a group of bars, specified in either of these format • Pixels (e.g. 50). • Percentage of the available width for each group (e.g. '20%'), no space between them. **Type:** number or string **Default:** The golden ratio (http://en.wikipedia.org/wiki/Golden_r
candlestick.hollowIsRising	If true, rising candles will appear hollow and falling candles will **Type:** boolean **Default:** false (will later be changed to true)

candlestick.fallingColor.fill	The fill color of falling candles, as an HTML color string. **Type:** string **Default:** auto (depends on the series color and hollowIsRising)
candlestick.fallingColor.stroke	The stroke color of falling candles, as an HTML color string. **Type:** string **Default:** auto (the series color)
candlestick.fallingColor.strokeWidth	The stroke width of falling candles, as an HTML color string. **Type:** 2 **Default:** number
candlestick.risingColor.fill	The fill color of rising candles, as an HTML color string. **Type:** string **Default:** auto (white or the series color, depending on hollowIsRi
candlestick.risingColor.stroke	The stroke color of rising candles, as an HTML color string. **Type:** string **Default:** auto (the series color or white, depending on hollowIsR
candlestick.risingColor.strokeWidth	The stroke width of rising candles, as an HTML color string. **Type:** number **Default:** 2
chartArea	An object with members to configure the placement and size of drawn, excluding axis and legends). Two formats are supported A simple number is a value in pixels; a number followed by % is `{left:20,top:0,width:'50%',height:'75%'}` **Type:** object **Default:** null
chartArea.backgroundColor	Chart area background color. When a string is used, it can be eit English color name. When an object is used, the following prope • **`stroke`**: the color, provided as a hex string or English color r • **`strokeWidth`**: if provided, draws a border around the chart 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 string, for example: `colors:['red','#004411']`. **Type:** Array of strings **Default:** default colors
crosshair	An object containing the crosshair (https://developers.google.c properties for the 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 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 histogram, bar, and column charts, this refers to the visible data rectangles in the others. In charts where *selecting data* creates this refers to the circles that appear upon hover or selection. Th and this option has no effect on other charts. (To change the op opacity (https://developers.google.com/chart/interactive/docs **Type:** number **Default:** 1.0
enableInteractivity	Whether the chart throws user-based events or reacts to user in throw 'select' or other interaction-based events (but *will* throw re hovertext or otherwise change depending on user input. **Type:** boolean **Default:** true
focusTarget	The type of the entity that receives focus on mouse hover. Also mouse click, and which data table element is associated with ev • 'datum' - Focus on a single data point. Correlates to a cell in • 'category' - Focus on a grouping of all data points along the r data table. In focusTarget 'category' the tooltip displays all the category val values of different series. **Type:** string **Default:** 'datum'
fontSize	The default font size, in pixels, of all text in the chart. You can o chart elements. **Type:** number **Default:** automatic

fontName	The default font face for all text in the chart. You can override th elements. **Type:** string **Default:** 'Arial'
forceIFrame	Draws the chart inside an inline frame. (Note that on IE8, this op in i-frames.) **Type:** boolean **Default:** false
hAxis	An object with members to configure various horizontal axis ele 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://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** automatic
hAxis.baselineColor	The color of the baseline for the horizontal axis. Can be any HTI `'#00cc00'`. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** 'black'
hAxis.direction	The direction in which the values along the horizontal axis grow 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 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 more details, see loading charts with a specific locale (https://developers.google.com/chart/interactive/docs/library_ . This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **Type:** string **Default:** auto
hAxis.gridlines	An object with members to configure the gridlines on the horizo object, you can use object literal notation, as shown here: `{color: '#333', count: 4}` This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **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 with chart computed gridlines. Allows formatting for years, mor 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://developers.google.com/chart/interactive/docs/datesa
	
	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://developers.google.com/chart/interactive/docs/custom **Type:** object **Default:** null
hAxis.minorGridlines.color	The color of the horizontal minor gridlines inside the chart area. **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 with chart computed minorGridlines. Allows formatting for year 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://developers.google.com/chart/interactive/docs/datesa
	
	Type: object
	Default: null

hAxis.logScale	**`hAxis`** property that makes the horizontal axis a logarithmic sc Set to true for yes. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **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 setting **`hAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero a negative number is the negative of the log of the absolute linear scale. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **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 **`fontSize`**.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-r size>}`**

hAxis.ticks	Replaces the automatically generated X-axis ticks with the spec should be either a valid tick value (such as a number, date, datet an object, it should have a **v** property for the tick value, and an o 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'},`** **`}`** This option is only supported for a **`continuous`** (https://developers.google.com/chart/interactive/docs/custom **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 `fontSize`. **Type:** object **Default:** `{color: 'black', fontName: <global-font-r size>}`
hAxis.allowContainerBoundaryTextCufoff	If false, will hide outermost labels rather than allow them to be c will allow label cropping. This option is only supported for a discrete (https://developers.google.com/chart/interactive/docs/custom **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 that this option is available only when the `hAxis.textPositi` This option is only supported for a discrete (https://developers.google.com/chart/interactive/docs/custom **Type:** boolean **Default:** automatic
hAxis.slantedTextAngle	The angle of the horizontal axis text, if it's drawn slanted. Ignore is in auto mode, and the chart decided to draw the text horizont This option is only supported for a discrete (https://developers.google.com/chart/interactive/docs/custom **Type:** number, 1—90 **Default:** 30
hAxis.maxAlternation	Maximum number of levels of horizontal axis text. If axis text la might shift neighboring labels up or down in order to fit labels c most number of levels to use; the server can use fewer levels, if This option is only supported for a discrete (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** 2
hAxis.maxTextLines	Maximum number of lines allowed for the text labels. Labels ca and the number of lines is, by default, limited by the height of th This option is only supported for a discrete (https://developers.google.com/chart/interactive/docs/custom **Type:** number

	Default: auto
hAxis.minTextSpacing	Minimum horizontal spacing, in pixels, allowed between two adj spaced too densely, or they are too long, the spacing can drop b of the label-unclutter measures will be applied (e.g, truncating th This option is only supported for a `discrete` (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** The value of **`hAxis.textStyle.fontSize`**
hAxis.showTextEvery	How many horizontal axis labels to show, where 1 means show label, and so on. Default is to try to show as many labels as pos This option is only supported for a `discrete` (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** automatic
hAxis.maxValue	Moves the max value of the horizontal axis to the specified valu Ignored if this is set to a value smaller than the maximum x-valu **`hAxis.viewWindow.max`** overrides this property. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** automatic
hAxis.minValue	Moves the min value of the horizontal axis to the specified value Ignored if this is set to a value greater than the minimum x-value **`hAxis.viewWindow.min`** overrides this property. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **Type:** number **Default:** automatic
hAxis.viewWindowMode	Specifies how to scale the horizontal axis to render the values w string values are supported: • 'pretty' - Scale the horizontal values so that the maximum an bit inside the left and right of the chart area. This will cause l **`haxis.viewWindow.max`** to be ignored. • 'maximized' - Scale the horizontal values so that the maximu left and right of the chart area. This will cause **`haxis.view`** **`haxis.viewWindow.max`** to be ignored.

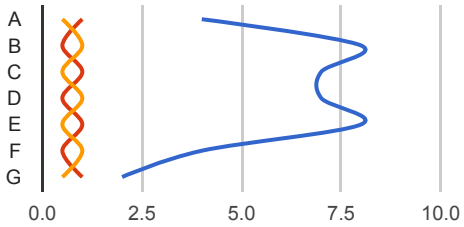
	• 'explicit' - A deprecated option for specifying the left and righ (Deprecated because it's redundant with **`haxis.viewWind` `haxis.viewWindow.max`**.) Data values outside these value **`hAxis.viewWindow`** object describing the maximum and m This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **Type:** string **Default:** Equivalent to 'pretty', but **`haxis.viewWindow.min`** an precedence if used.
hAxis.viewWindow	Specifies the cropping range of the horizontal axis. **Type:** object **Default:** null
hAxis.viewWindow.max	• For a `continuous` (https://developers.google.com/chart/interactive/docs/cus The maximum horizontal data value to render. • For a `discrete` (https://developers.google.com/chart/interactive/docs/cus The zero-based row index where the cropping window ends. be cropped out. In conjunction with **`vAxis.viewWindowMo`** [min, max) that denotes the element indices to display. In oth **`index < max`** will be displayed. Ignored when **`hAxis.viewWindowMode`** is 'pretty' or 'maximize **Type:** number **Default:** auto
hAxis.viewWindow.min	• For a `continuous` (https://developers.google.com/chart/interactive/docs/cus The minimum horizontal data value to render. • For a `discrete` (https://developers.google.com/chart/interactive/docs/cus The zero-based row index where the cropping window begin will be cropped out. In conjunction with **`vAxis.viewWindo`** range [min, max) that denotes the element indices to display **`min <= index < max`** will be displayed. 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 on neighboring points. If false, it will leave a break in the line at t This **is not** supported by Area (https://developers.google.com/chart/interactive/docs/gallery/ `isStacked: true/'percent'/'relative'/'absolute'` **Type:** boolean **Default:** false
isStacked	If set to true, series elements of the same type are stacked. Affe **Type:** boolean **Default:** false
legend	An object with members to configure various aspects of the leg 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 'right' legend, 'start' and 'end' are at the top and bottom, respecti would be at the left and right of the area, respectively. The default value depends on the legend's position. For 'bottom legends default to 'start'. **Type:** string **Default:** automatic
legend.maxLines	Maximum number of lines in the legend. Set this to a number gr legend. Note: The exact logic used to determine the actual num 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 the legend on the left, use the option **targetAxisIndex:** • '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 **fontSize**.
	Type: object
	Default: **{color: 'black', fontName: <global-font-n size>}**

lineDashStyle	The on-and-off pattern for dashed lines. For instance, **[4, 4]** w 4-length gaps, and **[5, 1, 3]** will repeat a 5-length dash, a 1-l gap, a 1-length dash, and a 3-length gap. See Dashed Lines (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 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', points documentation (https://developers.google.com/chart/in **Type:** string **Default:** 'circle'
pointSize	Diameter of displayed points in pixels. Use zero to hide all point series using the **`series`** property. If you're using a trendline (https://developers.google.com/chart/interactive/docs/gallery, affect the width of the trendline unless you override it with the **`t`** **Type:** number **Default:** 0
pointsVisible	Determines whether points will be displayed. Set to **`false`** to hi for individual series using the **`series`** property. If you're using a (https://developers.google.com/chart/interactive/docs/gallery, option will affect the visibility of the points on all trendlines unle **`trendlines.n.pointsVisible`** option. This can also be overridden using the style role (https://developers.google.com/chart/interactive/docs/roles#s **`{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://developers.google.com/chart/interactive/docs/custom (https://developers.google.com/chart/interactive/docs/custom **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 values for a series, specify an empty object {}. If a series or a va be used. Each object supports the following properties:
	
	• **annotations** - An object to be applied to annotations for th for instance, the **textStyle** for the series:
	
	`series: {`
	`  0: {`
	`  annotations: {`
	`  textStyle: {fontSize: 12, color: 'red'`
	`  }`
	`  }`
	`}`
	
	See the various **annotations** options for a more complete
	• **areaOpacity** - Overrides the global **areaOpacity** for this
	• **color** - The color to use for this series. Specify a valid HTM
	• **curveType** - Overrides the global **curveType** value for this
	• **fallingColor.fill** - Overrides the global **candlestick** series.
	• **fallingColor.stroke** - Overrides the global **candlesti** this series.
	• **fallingColor.strokeWidth** - Overrides the global **candlestick.fallingColor.strokeWidth** value for th
	• **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
	• **risingColor.fill** - Overrides the global **candlestick.** series.
	• **risingColor.stroke** - Overrides the global **candlestic** this series.
	• **risingColor.strokeWidth** - Overrides the global **candlestick.risingColor.strokeWidth** value for thi
	• **targetAxisIndex** - Which axis to assign this series to, wh opposite axis. Default value is 0; set to 1 to define a chart wh

	against different axes. At least one series much be allocated different scale for different axes.
	• **`type`** - The type of marker for this series. Valid values are 'lir 'steppedArea'. Note that bars are actually vertical bars (colur the chart's **`seriesType`** option.
	• **`visibleInLegend`** - A boolean value, where true means tha and false means that it should not. Default is true.
	
	You can specify either an array of objects, each of which applies can specify an object where each child has a numeric key indica example, the following two declarations are identical, and decla from the legend, and the fourth as red and absent from the lege
	
	`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: {}

seriesType	The default line type for any series not specified in the **`series`** 'area', 'bars', 'candlesticks', and 'steppedArea'. **Type:** string **Default:** 'line'

theme	A theme is a set of predefined option values that work together visual effect. Currently only one theme is available:
	• 'maximized' - Maximizes the area of the chart, and draws 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
	`fontSize`.
	Type: object
	Default: **`{color: 'black', fontName: <global-font-n`**
	`size>}`

tooltip	An object with members to configure various tooltip elements. T 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 **Note:** This only applies to HTML tooltips. If this is enabled with the chart bounds will be cropped. See Customizing Tooltip Cont (https://developers.google.com/chart/interactive/docs/custom **Type:** boolean **Default:** false
tooltip.isHtml	If set to true, use HTML-rendered (rather than SVG-rendered) too (https://developers.google.com/chart/interactive/docs/custom

	Note: customization of the HTML tooltip content via the tooltip (https://developers.google.com/chart/interactive/docs/roles#t Bubble Chart (https://developers.google.com/chart/interactive/ **Type:** boolean **Default:** false
tooltip.showColorCode	If true, show colored squares next to the series information in th **`focusTarget`** is set to 'category', otherwise the default is false **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 **`fontSize`**.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-r 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'

vAxes	Specifies properties for individual vertical axes, if the chart has is a **`vAxis`** object, and can contain all the properties supported 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`** series 2 to the right axis and specifies a custom title and text st
	
	`{`
	`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 of objects, one per axis. For example, the following array-style n shown above:
	
	vAxes: [
	{}, // Nothing specified for axis 0
	{
	title:'Losses',
	textStyle: {color: 'red'} // Axis 1
	}
	]
	
	Type: Array of object, or object with child objects
	Default: null

vAxis	An object with members to configure various vertical axis eleme you can use object literal notation, as shown here: {title: 'Hello', titleTextStyle: {color: '#FF00 **Type:** object **Default:** null
vAxis.baseline	**`vAxis`** property that specifies the baseline for the vertical axis. grid line or smaller than the lowest grid line, it will be rounded to **Type:** number **Default:** automatic
vAxis.baselineColor	Specifies the color of the baseline for the vertical axis. Can be a **`'red'`** or **`'#00cc00'`**. **Type:** number

	Default: 'black'
vAxis.direction	The direction in which the values along the vertical axis grow. S values. **Type:** 1 or -1 **Default:** 1
vAxis.format	A format string for numeric axis labels. This is a subset of the I (http://icu-project.org/apiref/icu4c/classDecimalFormat.html# `{format:'#,###%'}` will display values "1,000%", "750%", and 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., 8 The actual formatting applied to the label is derived from the lo more details, see loading charts with a specific locale (https://developers.google.com/chart/interactive/docs/library_ . **Type:** string **Default:** auto
vAxis.gridlines	An object with members to configure the gridlines on the vertica 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 compute the number of gridlines. **Type:** number **Default:** 5

vAxis.gridlines.units	Overrides the default format for various aspects of date/datetin with chart computed gridlines. Allows formatting for years, mor 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://developers.google.com/chart/interactive/docs/datesa **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 with chart computed minorGridlines. Allows formatting for year seconds, and milliseconds. General format is: `gridlines: { units: { years: {format: [/*format strings here*/]},`

	<pre>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
	}
	}</pre>
	Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesa
	
	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 p setting **`vAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero a negative number is the negative of the log of the absolute linear scale. This option is only supported for a `continuous` (https://developers.google.com/chart/interactive/docs/custom **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
	
	<pre>{ color: <string>,
	fontName: <string>,
	fontSize: <number>,
	bold: <boolean>,
	italic: <boolean> }</pre>

	The **color** can be any HTML color string, for example: '**red**' c **fontSize**. **Type:** object **Default:** {**color: 'black', fontName: <global-font-r size>**}
vAxis.ticks	Replaces the automatically generated Y-axis ticks with the spec should be either a valid tick value (such as a number, date, datet an object, it should have a **v** property for the tick value, and an o 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'}, }** **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 **fontSize**.
	
	Type: object
	Default: {**color: 'black', fontName: <global-font-r size>**}

vAxis.maxValue	Moves the max value of the vertical axis to the specified value; t Ignored if this is set to a value smaller than the maximum y-valu **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 Ignored if this is set to a value greater than the minimum y-valu **`vAxis.viewWindow.min`** overrides this property. **Type:** number **Default:** null
vAxis.viewWindowMode	Specifies how to scale the vertical axis to render the values with values are supported: • 'pretty' - Scale the vertical values so that the maximum and m inside the top and bottom of the chart area. This will cause **v** **`vaxis.viewWindow.max`** to be ignored. • 'maximized' - Scale the vertical values so that the maximum and bottom of the chart area. This will cause **`vaxis.viewW`** **`vaxis.viewWindow.max`** to be ignored. • 'explicit' - A deprecated option for specifying the top and bot (Deprecated because it's redundant with **`vaxis.viewWind`** **`vaxis.viewWindow.max`**. Data values outside these value **`vAxis.viewWindow`** object describing the maximum and m **Type:** string **Default:** Equivalent to 'pretty', but **`vaxis.viewWindow.min`** an 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 (#Events)event is fired. Extended description (https://developers.google.com/chart/interactive/docs/reference#vis **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
	The format for `id` isn't yet documented (they're the return values of eve
	(https://developers.google.com/chart/interactive/docs/events)), but h
	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) c
	
	`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
	
	Return Type: object

`getChartAreaBoundingBox()`	Returns an object containing the left, top, width, and height of the chart (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
	
	Return Type: object

`getChartLayoutInterface()`	Returns an object containing information about the onscreen placemen 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 container's left edge. Can be negative. Example: `chart.getChartLayoutInterface().getHAxisValue` 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://developers.google.com/chart/interactive/docs/printing). **Return Type:** string

`getSelection()`	Returns an array of the selected chart entities. Selectable entities are p steps, annotations, legend entries and categories . A point, bar, step, or corresponds to a cell in the data table, a legend entry to a column (row null), and a category to a row (column index is null). For this chart, only can be selected at any given moment. Extended description (https://developers.google.com/chart/interactive/docs/reference#vis . **Return Type:** Array of selection elements
`getVAxisValue(position, optional_axis_index)`	Returns the logical vertical value at **`position`**, which is an offset from container's top edge. Can be negative. Example: **`chart.getChartLayoutInterface().getVAxisValue`** 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 co Example: **`chart.getChartLayoutInterface().getXLocation(`** 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 co Example: **`chart.getChartLayoutInterface().getYLocation(`** 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 should specify 3 properties: **`id`**— the ID of the action being set, **`text`** – should appear in the tooltip for the action, and **`action`** — the function 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://developers.google.com/chart/interactive/docs/reference#vis **Return Type: none**

`setSelection()`	Selects the specified chart entities. Cancels any previous selection. Se entities are points, bars, steps, annotations, legend entries and categor bar, step, or annotation corresponds to a cell in the data table, a legenc column (row index is null), and a category to a row (column index is nu chart, only one entity can be selected at a time. `Extended descript` (https://developers.google.com/chart/interactive/docs/reference#vis . **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://developers.google.com/chart/interactive/docs/basic_interactivity), Handling Events (https://developers.google.com/chart/interactive/docs/events), and Firing Events (https://developers.google.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