Tooltips

Tooltips: an introduction

Tooltips are the little boxes that pop up when you hover over something. (Hovercards are more general, and can appear anywhere on the screen; tooltips are always attached to something, like a dot on a scatter chart, or a bar on a bar chart.)

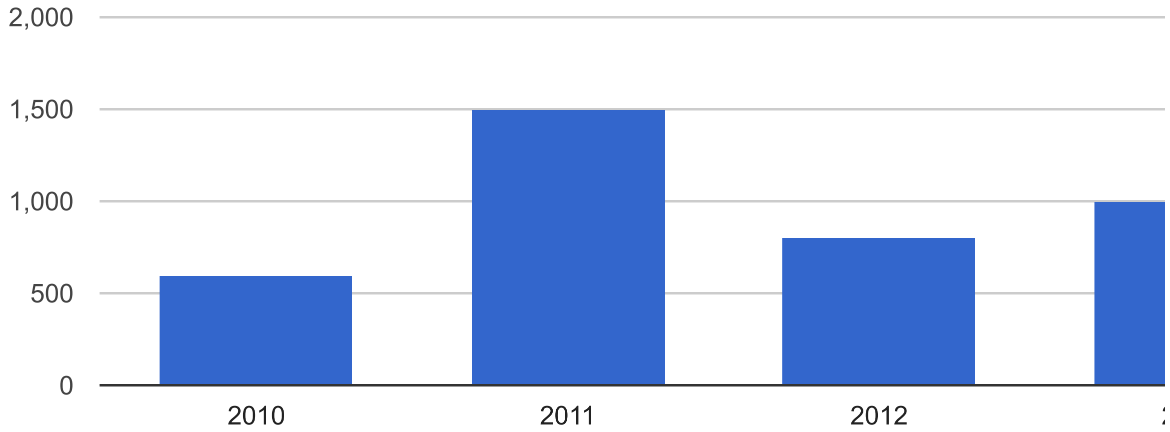
In this documentation, you'll learn how to create and customize tooltips in Google Charts.

Specifying the tooltip type

Google Charts automatically creates tooltips for all core charts. They'll be rendered as SVG by default, except under IE 8 where they'll be rendered as VML. You can create HTML tooltips (#enabling_html_tooltip) on core charts by specifying `tooltip.isHtml: true` in the chart options (https://google-developers.appspot.com/chart/interactive/docs/customizing_charts) passed to the draw() (https://google-developers.appspot.com/chart/interactive/docs/drawing_charts#chart.draw) call, which will also allow you to create Tooltip Actions (#tooltip_actions).

Standard tooltips

In the absence of any custom content, the tooltip content is automatically generated based on the underlying data. You can see the tooltip by hovering your mouse over any of the bars in the chart.



Hover over the bars to see standard tooltips.

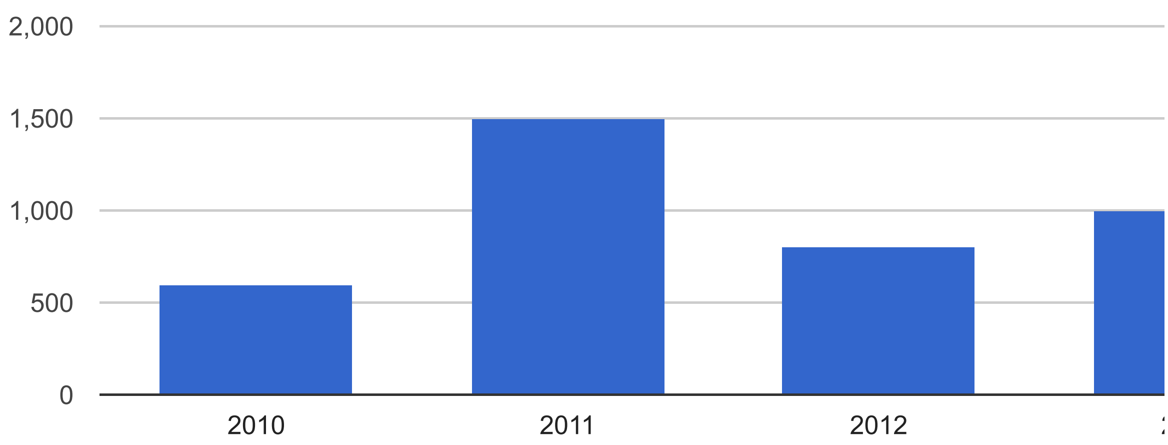
Customizing tooltip content

In this example, we add custom content to the tooltips by adding a new column to the DataTable (https://google-developers.appspot.com/chart/interactive/docs/datatables_dataviews) and marking it with the tooltip role (https://google-developers.appspot.com/chart/interactive/docs/roles#tooltiprole).

Note: this is **not** supported by the Bubble Chart (https://google-developers.appspot.com/chart/interactive/docs/gallery/bubblechart) visualization.

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

    function drawChart() {
      var dataTable = new google.visualization.DataTable();
      dataTable.addColumn('string', 'Year');
      dataTable.addColumn('number', 'Sales');
      // A column for custom tooltip content
      dataTable.addColumn({type: 'string', role: 'tooltip'});
      dataTable.addRows([
        ['2010', 600,'$600K in our first year!'],
        ['2011', 1500, 'Sunspot activity made this our best year ever!'],
        ['2012', 800, '$800K in 2012.'],
        ['2013', 1000, '$1M in sales last year.']
      ]);
```

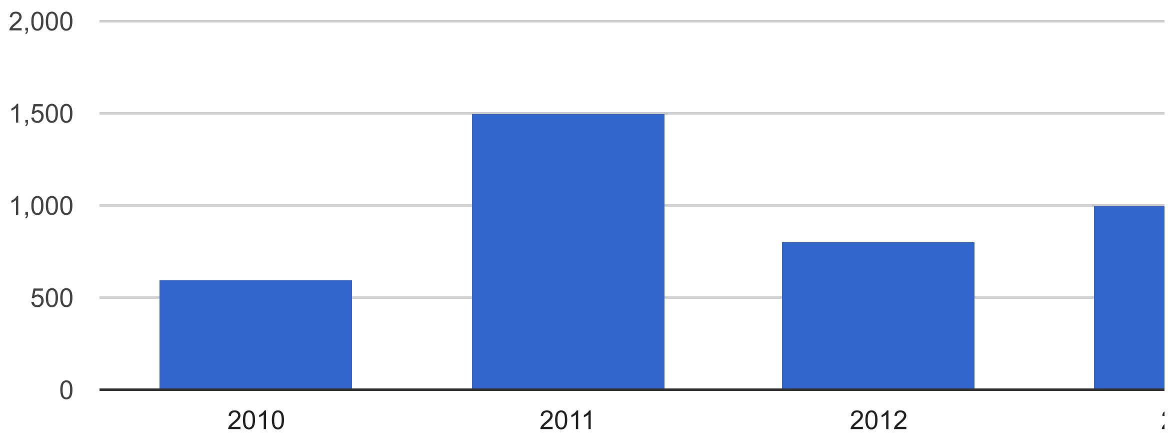
```
        var options = { legend: 'none' };
        var chart = new google.visualization.ColumnChart(document.getElementBy
        chart.draw(dataTable, options);
      }
```



Using HTML tooltips

This example builds on the previous one by enabling HTML tooltips. Note the addition of **`tooltip.isHtml: true`** to the chart options.

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

      function drawChart() {
        var dataTable = new google.visualization.DataTable();
        dataTable.addColumn('string', 'Year');
        dataTable.addColumn('number', 'Sales');
        // A column for custom tooltip content
        dataTable.addColumn({type: 'string', role: 'tooltip'});
        dataTable.addRows([
          ['2010', 600,'$600K in our first year!'],
          ['2011', 1500, 'Sunspot activity made this our best year ever!'],
          ['2012', 800, '$800K in 2012.'],
          ['2013', 1000, '$1M in sales last year.']
        ]);

        var options = {
          tooltip: {isHtml: true},
          legend: 'none'
        };
        var chart = new google.visualization.ColumnChart(document.getElementBy
        chart.draw(dataTable, options);
      }
```



That doesn't look much different, but now we can customize the HTML.

Customizing HTML content

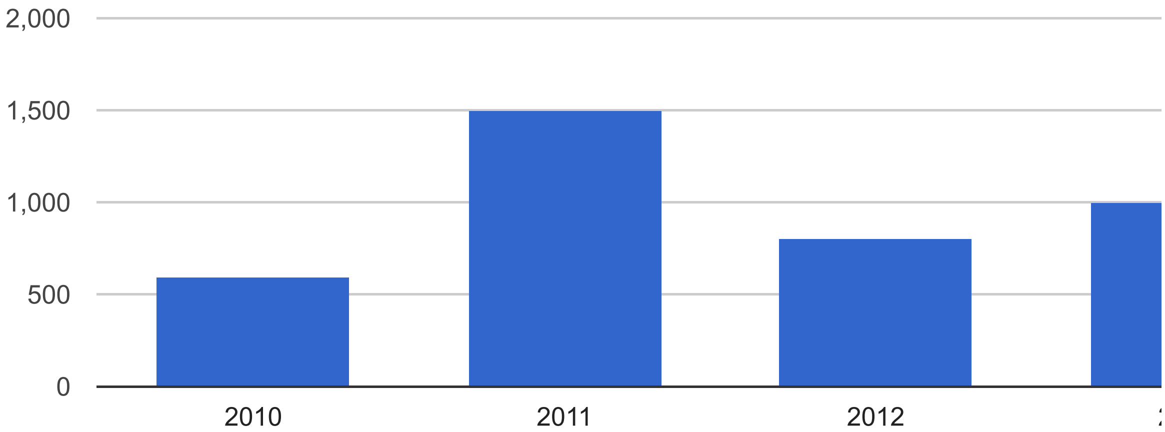
The previous examples have all used tooltips with plain text content, but the real power of HTML tooltips comes through when you can customize them using HTML markup. To enable this, you must do two things:

1. Specify `tooltip.isHtml: true` in the chart options. This tells the chart to draw the tooltips in HTML (as opposed to SVG).
2. Mark an entire column, or a specific cell, in the data table with the `html` custom property. A datatable column with the tooltip role and HTML property would be: `dataTable.addColumn({'type': 'string', 'role': 'tooltip', 'p': {'html': true}})`

Note: this does **not** work with the Bubble Chart (https://google-developers.appspot.com/chart/interactive/docs/gallery/bubblechart) visualization. The content of Bubble Chart HTML tooltips is not customizable.

Important: Make sure that the HTML in your tooltips comes from a trusted source. If the custom HTML content contains any user generated content, escaping that content is vital. Otherwise, your beautiful visualizations may be open to XSS (https://en.wikipedia.org/wiki/Cross-site_scripting) attacks.

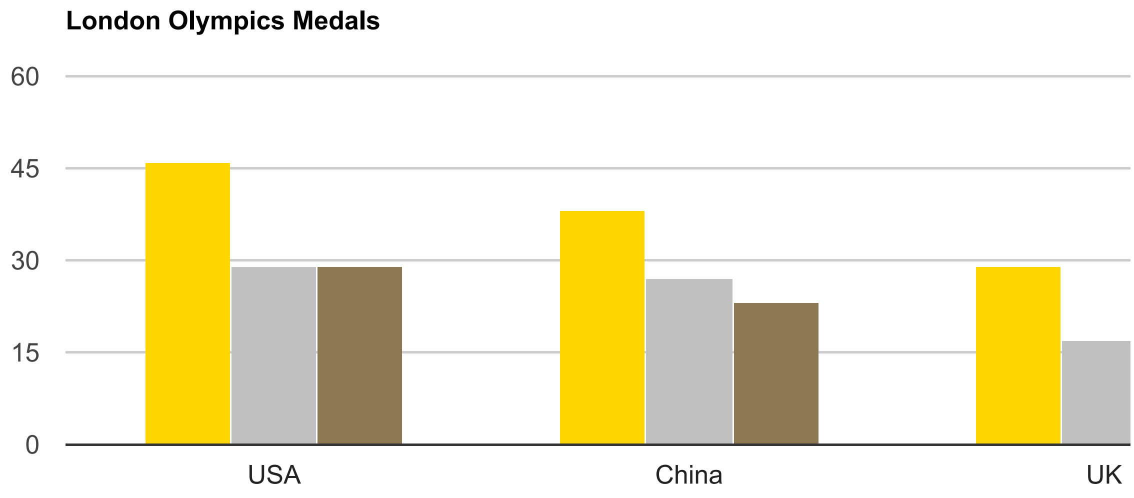
Custom HTML content can be as simple as adding an `<img>` tag or bolding some text:



Custom HTML content can also include dynamically generated content. Here, we add a tooltip containing a dynamically generated table for each category value. Since the tooltip is attached to the row value, hovering over any of the bars will display the HTML tooltip.

This example demonstrates how a custom HTML tooltip can be attached to a domain column. (In previous examples, it was attached to a data column.) To turn on the tooltip for the domain axis, set the `focusTarget: 'category'` option.

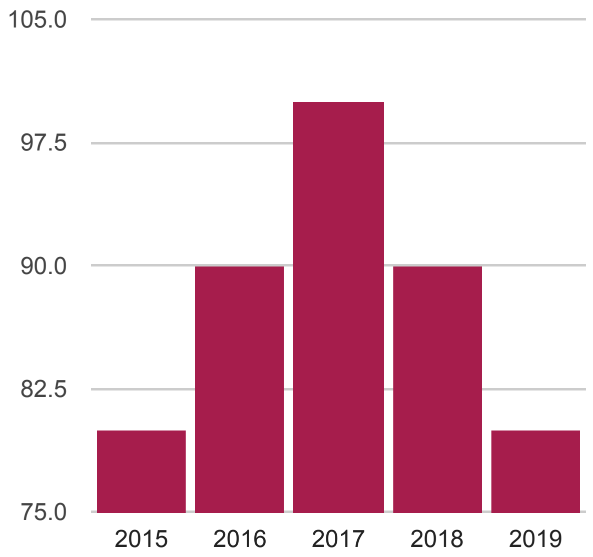
```
function drawChart() {
  var dataTable = new google.visualization.DataTable();
  dataTable.addColumn('string', 'Country');
  // Use custom HTML content for the domain tooltip.
  dataTable.addColumn({'type': 'string', 'role': 'tooltip', 'p': {'html': tru
  dataTable.addColumn('number', 'Gold');
  dataTable.addColumn('number', 'Silver');
  dataTable.addColumn('number', 'Bronze');

  dataTable.addRows([
    ['USA', createCustomHTMLContent('https://upload.wikimedia.org/wikipedia/c
    ['China', createCustomHTMLContent('https://upload.wikimedia.org/wikipedia
    ['UK', createCustomHTMLContent('https://upload.wikimedia.org/wikipedia/co
  ]);
  var options = {
    title: 'London Olympics Medals',
    colors: ['#FFD700', '#C0C0C0', '#8C7853'],
    // This line makes the entire category's tooltip active.
    focusTarget: 'category',
    // Use an HTML tooltip.
    tooltip: { isHtml: true }
  };

  // Create and draw the visualization.
  new google.visualization.ColumnChart(document.getElementById('custom_html_co
}

function createCustomHTMLContent(flagURL, totalGold, totalSilver, totalBronze
  return '<div style="padding:5px 5px 5px 5px;">' +
      '<img src="' + flagURL + '" style="width:75px;height:50px"><br/>' +
      '<table class="medals_layout">' + '<tr>' +
      '<td><img src="https://upload.wikimedia.org/wikipedia/commons/1/15/Gold_
      '<td><b>' + totalGold + '</b></td>' + '</tr>' + '<tr>' +
      '<td><img src="https://upload.wikimedia.org/wikipedia/commons/0/03/Silve
      '<td><b>' + totalSilver + '</b></td>' + '</tr>' + '<tr>' +
      '<td><img src="https://upload.wikimedia.org/wikipedia/commons/5/52/Bronz
      '<td><b>' + totalBronze + '</b></td>' + '</tr>' + '</table>' + '</div>'
}
```



Rotating tooltips

Tooltips in Google Charts can be rotated with CSS. In the chart below, the tooltips are rotated 30° clockwise using this inline CSS immediately before the chart div:

`<style>div.google-visualization-tooltip { transform: rotate(30deg); } </style>`



CODE IT YOURSELF ON JSFIDDLE

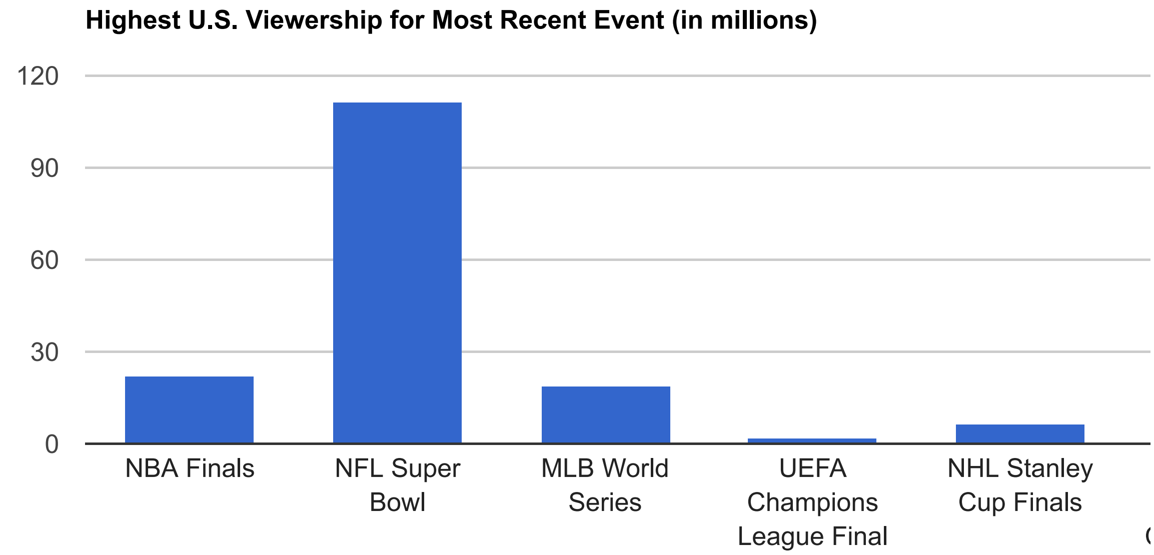
Note that this will only work for HTML tooltips, i.e., those with the option `isHtml: true`.

```
<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', 'Fixations'],
          ['2015',  80],
          ['2016',  90],
          ['2017',  100],
          ['2018',  90],
          ['2019',  80],
        ]);
        var options = {
          tooltip: { isHtml: true },    // CSS styling affects only HTML tool
          legend: { position: 'none' },
          bar: { groupWidth: '90%' },
          colors: ['#A61D4C']
        };

        var chart = new google.visualization.ColumnChart(document.getElementB

        chart.draw(data, options);
      }
    </script>
  </head>
  <body>
    <!-- The next line rotates HTML tooltips by 30 degrees clockwise. -->
    <style>div.google-visualization-tooltip { transform: rotate(30deg); }</st
    <div id="tooltip_rotated" style="width: 400px; height: 400px;"></div>
  </body>
</html>
```

Placing charts in tooltips

HTML tooltips can include most any HTML content you like—even a Google Chart. With charts inside tooltips, your users can get additional information when they hover over a data element: a good way to provide high level detail at first glance while letting people dive deeper when they like.

The column chart below shows a chart of the highest recent viewership of several major sporting events, with the tooltips for each showing a line chart of the average viewership over the last ten years.



A couple of things to note here. First, a printable chart (https://google-developers.appspot.com/chart/interactive/docs/printing) needs to be drawn for each set of data to be shown in a tooltip. Second, each tooltip chart needs a "ready" event handler (https://google-developers.appspot.com/chart/interactive/docs/events#The_Ready_Event), which we call via `addListener` to create a printable chart.

IMPORTANT: *All* tooltip charts *must* be drawn before the primary chart. This is necessary in order to grab the images to add to the primary chart's DataTable.

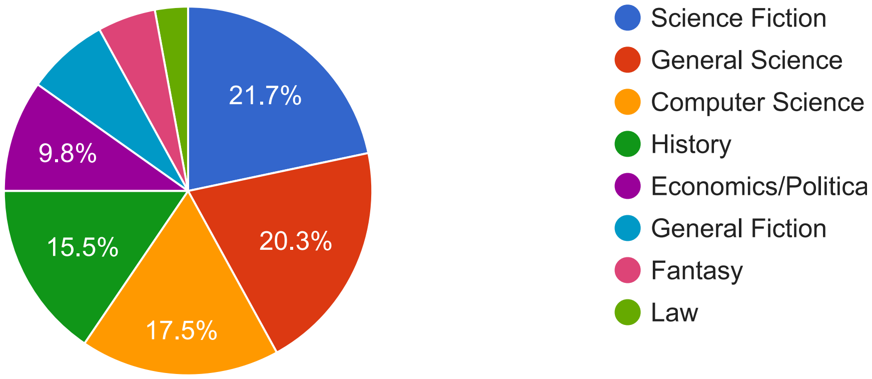
IMPORTANT PARTS   FULL WEB PAGE

```
// Draws your charts to pull the PNGs for your tooltips.
function drawTooltipCharts() {

  var data = new google.visualization.arrayToDataTable(tooltipData);
  var view = new google.visualization.DataView(data);

  // For each row of primary data, draw a chart of its tooltip data.
  for (var i = 0; i < primaryData.length; i++) {

    // Set the view for each event's data
    view.setColumns([0, i + 1]);

    var hiddenDiv = document.getElementById('hidden_div');
    var tooltipChart = new google.visualization.LineChart(hiddenDiv);

    google.visualization.events.addListener(tooltipChart, 'ready', fun

      // Get the PNG of the chart and set is as the src of an img tag.
```

```
            var tooltipImg = '<img src="' + tooltipChart.getImageURI() + '">

            // Add the new tooltip image to your data rows.
            primaryData[i][2] = tooltipImg;

          });
          tooltipChart.draw(view, tooltipOptions);
        }
        drawPrimaryChart();
      }

      function drawPrimaryChart() {

        var data = new google.visualization.DataTable();
        data.addColumn('string', 'Event');
        data.addColumn('number', 'Highest Recent Viewership');

        // Add a new column for your tooltips.
        data.addColumn({
          type: 'string',
          label: 'Tooltip Chart',
          role: 'tooltip',
          'p': {'html': true}
        });

        // Add your data (with the newly added tooltipImg).
        data.addRows(primaryData);

        var visibleDiv = document.getElementById('visible_div');
        var primaryChart = new google.visualization.ColumnChart(visibleDiv);
        primaryChart.draw(data, primaryOptions);

      }
```

Tooltip actions

Tooltips can also include JavaScript-defined actions. As a simple example, here's a tooltip with an action that pops up a dialog box when the user clicks on "See sample book":



The `tooltip` option triggers when the user selects a wedge of the pie, causing the code defined in `setAction` to be run:

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

function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Genre', 'Percentage of my books'],
    ['Science Fiction', 217],
    ['General Science', 203],
    ['Computer Science', 175],
    ['History', 155],
    ['Economics/Political Science', 98],
    ['General Fiction', 72],
    ['Fantasy', 51],
    ['Law', 29]
  ]);

  var chart = new google.visualization.PieChart(
    document.getElementById('tooltip_action'));

  var options = { tooltip: { trigger: 'selection' }};

  chart.setAction({
    id: 'sample',
    text: 'See sample book',
    action: function() {
      selection = chart.getSelection();
      switch (selection[0].row) {
        case 0: alert('Ender\'s Game'); break;
        case 1: alert('Feynman Lectures on Physics'); break;
```

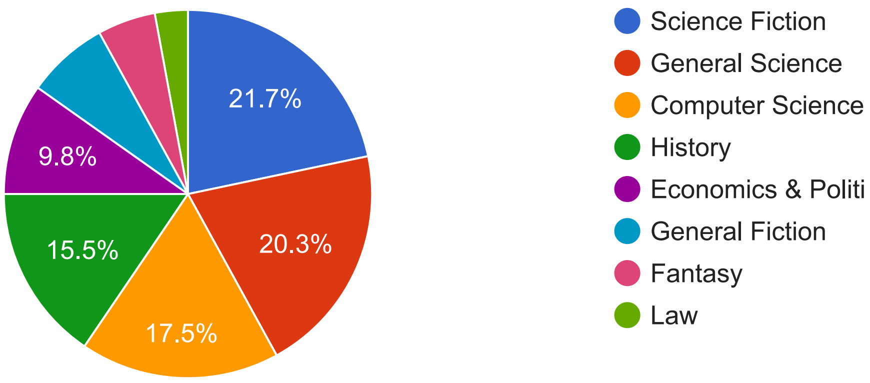
```
              case 2: alert('Numerical Recipes in JavaScript'); break;
              case 3: alert('Truman'); break;
              case 4: alert('Freakonomics'); break;
              case 5: alert('The Mezzanine'); break;
              case 6: alert('The Color of Magic'); break;
              case 7: alert('The Law of Superheroes'); break;
            }
          }
        });

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

Actions can be `visible`, `enabled`, both, or neither. When a Google Chart is rendered, a tooltip action is only shown if it's visible, and only clickable if it's enabled. (Disabled but visible actions are greyed out.)

`visible` and `enabled` should be functions that return true or false values. Those functions can depend on the element ID and user selection, allowing you to fine-tune action visibility and clickability.

Tooltips can trigger on `focus` as well as `selection`. However, with tooltip actions, `selection` is preferable. That causes the tooltip to persist, allowing the user to select the action more easily.

The actions needn't be alerts, of course: anything you can do from JavaScript, you can do from an action. Here, we'll add two actions: one to enlarge a wedge of our pie chart, and another to shrink it.

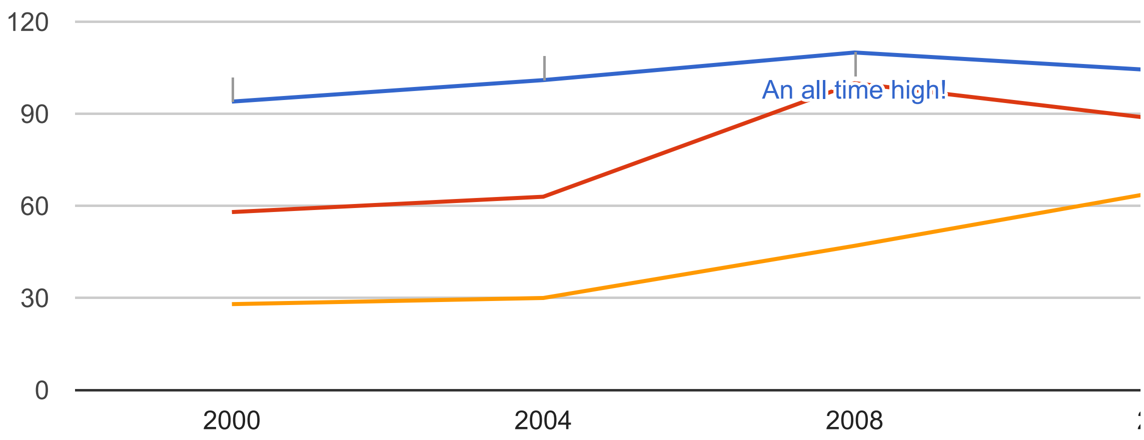


```
        google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Genre', 'Percentage of my books'],
          ['Science Fiction', 217],
          ['General Science', 203],
          ['Computer Science', 175],
          ['History', 155],
          ['Economics & Political Science', 98],
          ['General Fiction', 72],
          ['Fantasy', 51],
          ['Law', 29]
        ]);

        var chart = new google.visualization.PieChart(
          document.getElementById('tooltip_action_2'));

        var options = { tooltip: { trigger: 'selection' }};

        chart.setAction({
          id: 'increase',
          text: 'Read 20 more books',
          action: function() {
            data.setCell(chart.getSelection()[0].row, 1,
                         data.getValue(chart.getSelection()[0].row, 1) + 20);
            chart.draw(data, options);
          }
        });

        chart.setAction({
          id: 'decrease',
          text: 'Read 20 fewer books',
          action: function() {
            data.setCell(chart.getSelection()[0].row, 1,
                         data.getValue(chart.getSelection()[0].row, 1) - 20);
            chart.draw(data, options);
          }
        });

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

Annotating data

You can overlay text directly onto a Google Chart by using `annotationText` instead of `tooltip` as the column role (https://google-developers.appspot.com/chart/interactive/docs/roles#whatrolesavailable). (You can see the tooltip by hovering over the annotation with your mouse.)

```
function drawChart() {
  var dataTable = new google.visualization.DataTable();
  dataTable.addColumn('string', 'Year');
  dataTable.addColumn('number', 'USA');
  dataTable.addColumn({type: 'string', role: 'annotation'});
  dataTable.addColumn({type: 'string', role: 'annotationText', p: {html:true}
  dataTable.addColumn('number', 'China');
  dataTable.addColumn('number', 'UK');
  dataTable.addRows([
    ['2000',  94, '',  '', 58, 28],
    ['2004', 101, '',  '', 63, 30],
    ['2008', 110, 'An all time high!', '<img width=100px src="https://upload.
    ['2012', 104, '',  '', 88, 65]
  ]);

  var options = { tooltip: {isHtml: true}};
  var chart = new google.visualization.LineChart(document.getElementById('tt_
  chart.draw(dataTable, options);
}
```



Supported charts

HTML tooltips are currently supported for the following chart types:

- AreaChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/areachart)
- BarChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/barchart)
- CalendarChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/calendar)
- CandlestickChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/candlestickchart)
- ColumnChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/columnchart)
- ComboChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/combochart)
- LineChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/linechart)
- PieChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/piechart)
- Sankey Diagrams (https://google-developers.appspot.com/chart/interactive/docs/gallery/sankey)
- ScatterChart (https://google-developers.appspot.com/chart/interactive/docs/gallery/scatterchart)
- Timeline (https://google-developers.appspot.com/chart/interactive/docs/gallery/timeline)

If you are using the **annotationText** or **tooltip** roles, please see the documentation on roles (https://google-developers.appspot.com/chart/interactive/docs/roles#changes_deprecations) to read about future changes and how to avoid future pain.

上次更新日期：二月 23, 2017