

IDV460

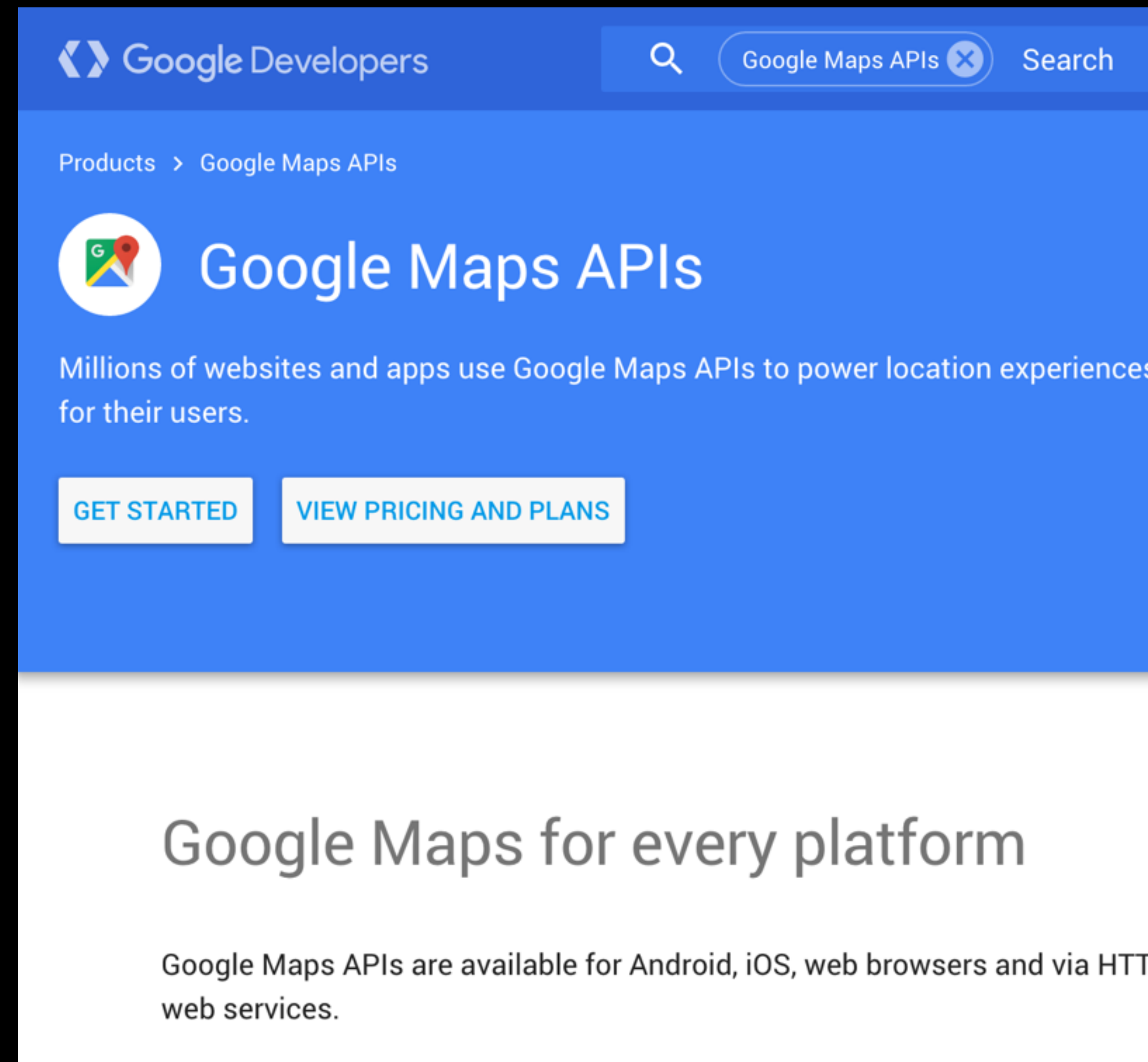
INTERACTIVE DATA VIZ SPRING 16

MAPS

WHAT IS HIGHMAPS?

Highcharts is an **API**, which stands for application programming interface. An API is a set of routines, protocols, and tools for building software and applications.

A good API makes it easier to develop a program by providing all the building blocks, which are then put together by the programmer. Facebook, Google Maps and YouTube are examples of APIs, which allow you to embed content into your own page, often with a few lines of code.

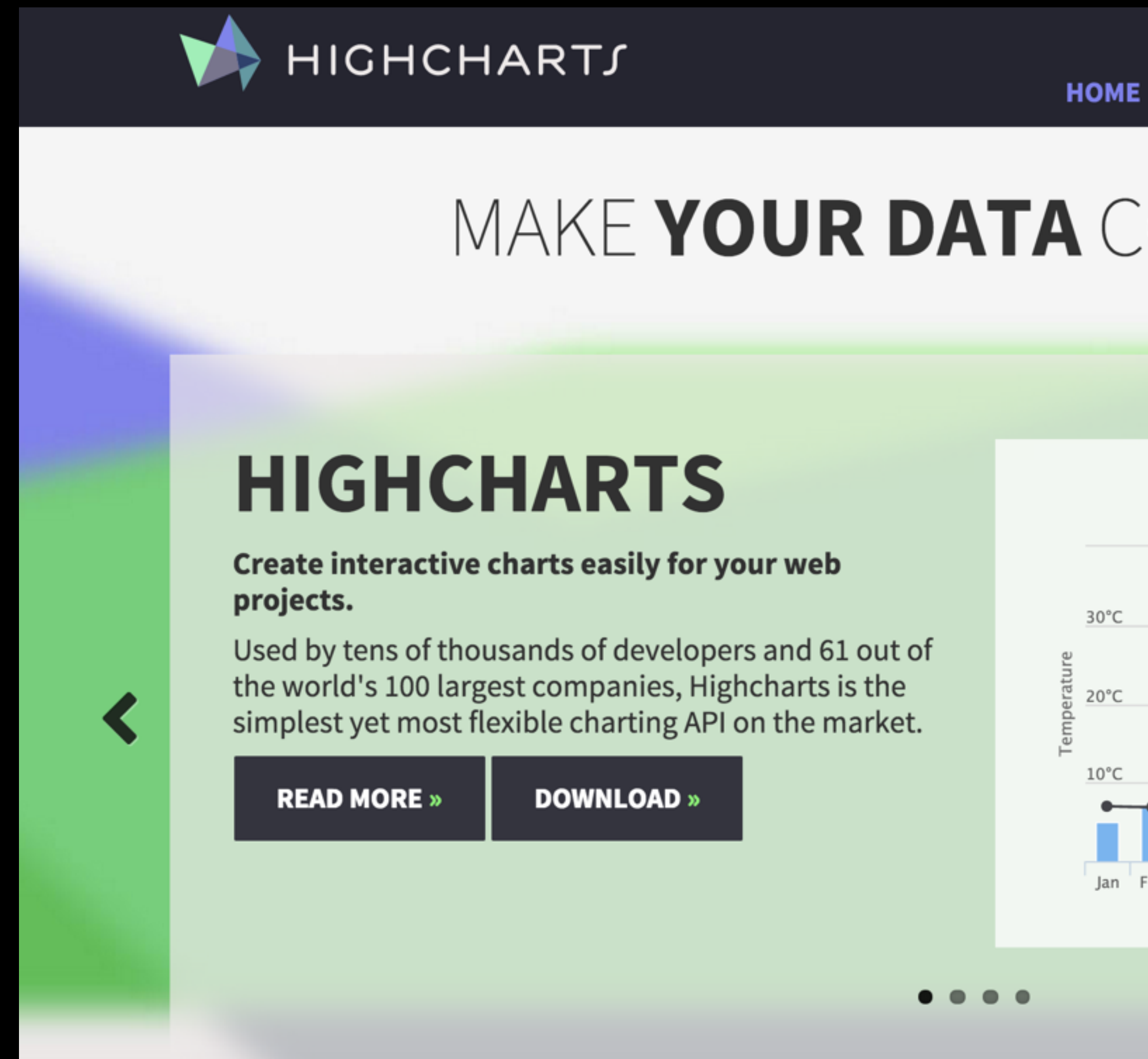


The image shows a screenshot of the Google Developers website for Google Maps APIs. The page has a blue header with the Google Developers logo and a search bar containing "Google Maps APIs". Below the header, there is a breadcrumb trail "Products > Google Maps APIs" and a large heading "Google Maps APIs" next to a circular icon featuring a map and a location pin. A sub-heading states "Millions of websites and apps use Google Maps APIs to power location experiences for their users." Two prominent buttons are visible: "GET STARTED" and "VIEW PRICING AND PLANS". The main content area below the blue header is white and features the heading "Google Maps for every platform" and a paragraph stating "Google Maps APIs are available for Android, iOS, web browsers and via HTTP web services."

WHAT IS HIGHCHARTS?

Highcharts is a charting library written in pure JavaScript, offering an easy way of adding interactive charts to your web site or web application.

Highcharts currently supports line, spline, area, areaspline, column, bar, pie, scatter, angular gauges, arearange, areasplinerange, columnrange, bubble, box plot, error bars, funnel, waterfall and polar chart types.



The image shows a screenshot of the Highcharts website homepage. At the top, there is a dark blue header with the Highcharts logo (a stylized 'H' made of three triangles) and the word 'HIGHCHARTS' in white. To the right of the logo is a 'HOME' link. Below the header, the main content area has a light green background. The headline reads 'MAKE YOUR DATA C'. Below this, the word 'HIGHCHARTS' is written in large, bold, black letters. Underneath, a sub-headline says 'Create interactive charts easily for your web projects.' followed by a paragraph: 'Used by tens of thousands of developers and 61 out of the world's 100 largest companies, Highcharts is the simplest yet most flexible charting API on the market.' There are two dark blue buttons with white text: 'READ MORE »' and 'DOWNLOAD »'. On the right side, there is a small preview of a temperature chart with a line and bars, showing a temperature scale from 10°C to 30°C and labels for 'Jan' and 'F'. At the bottom right, there are four small black dots, likely a navigation indicator.

LOADING THE NECESSARY JAVASCRIPT FILES

Highcharts requires two Javascript files to run: **highcharts.js** and **jQuery**.

We will be learning more about jQuery – a powerful Javascript library – after the break, but for now, keep in mind that it is needed by Highcharts and it needs to precede the highcharts.js link in your <head> tag.

```
/*! jQuery v2.1.4 | (c) 2005, 2015 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){"object"===typeof module&&"object"===typeof
module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new
Error("jQuery requires a window with a document");return b(a)}:b(a)}("undefined"!==typeof
window?window:this,function(a,b){var
c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.hasOwnProperty,k={},l
=a.document,m="2.1.4",n=function(a,b){return new
n.fn.init(a,b)},o=/^\s\uFEFF\uA0+|[\s\uFEFF\uA0]+$/g,p=/^-ms-/,q=/-([\da-z])/gi,r=
function(a,b){return
b.toUpperCase() /*
function(){ret Highcharts JS v4.2.3 (2016-02-08)
null!=a?0>a?th (c) 2009-2016 Torstein Honsi
b=n.merge(this License: www.highcharts.com/license
b.prevObject=t */
n.each(this,a, (function(E,X){typeof
a.call(b,c,b)} this.pushStack
this.eq(0)},la module==="object"&&module.exports?module.exports=E.document?X(E):X:E.Highcharts=X(E))
b=this.length, typeof window!=="undefined"?window:this,function(E){function X(a,b){var c="Highcharts
this.pushStack error #"+a+": www.highcharts.com/errors/"+a;if(b)throw
this.prevObject Error(c);E.console&&console.log(c)}function
.extend=functi pb(a,b,c){this.options=b;this.elem=a;this.prop=c}function D(){var
a,b,c,d,e,f,g= a,b=arguments,c,d={},e=function(a,b){var c,d;typeof a!=="object"&&(a={});for(d in
g&&(j=g,g=argu b)b.hasOwnProperty(d)&&(c=b[d],a[d]=c&&typeof c==="object"&&
g|n.isFunctio Object.prototype.toString.call(c)!=="[object Array]"&&d!=="renderTo"&&typeof
a)c.nodeType!=="number"?e(a[d]||{},c):b[d]);return
a};b[0]===!0&&(d=b[1],b=Array.prototype.slice.call(b,2));c=b.length;for(a=0;a<c;a++)d=e(d,
0!=="d&&(g[b]=d b[a]);return d}function C(a,b){return parseInt(a,b||10)}function xa(a){return typeof
g},n.extend({e a==="string"}function Y(a){return a&&typeof a==="object"}function Ia(a){return
g},n.extend({e Object.prototype.toString.call(a)=="[object Array]"}function ma(a){return typeof
function(a){th a==="number"}function Da(a){return W.log(a)/
Error(a)},noop W.LN10}function na(a){return W.pow(10,a)}function oa(a,b){for(var
Array.isArray, c=a.length;c--;)if(a[c]===b){a.splice(c,1);break}}function q(a){return
null!=a&&a===a a!==z&&a!==null}function K(a,b,c){var
isPlainObject: d,e;if(xa(b))q(c)?a.setAttribute(b,c):a&&a.getAttribute&&(e=a.getAttribute(b));else
constructor&& if(q(b)&&Y(b))for(d in b)a.setAttribute(d,b[d]);return e}function ta(a){return
function(a){va Ia(a)?a:[a]}function Pa(a,b,c){if(b)return setTimeout(a,b,c);a.call(0,c)}function
null==a?+"":" l(a,b){if(va&&!ca&&b&&b.opacity!==z)h.filter="alpha(opacity="+b.opacity*100+")"::u(a style
a}.globalEval;
```

LOADING THE NECESSARY JAVASCRIPT FILES

You can download the latest Highcharts code from

<http://www.highcharts.com/download>

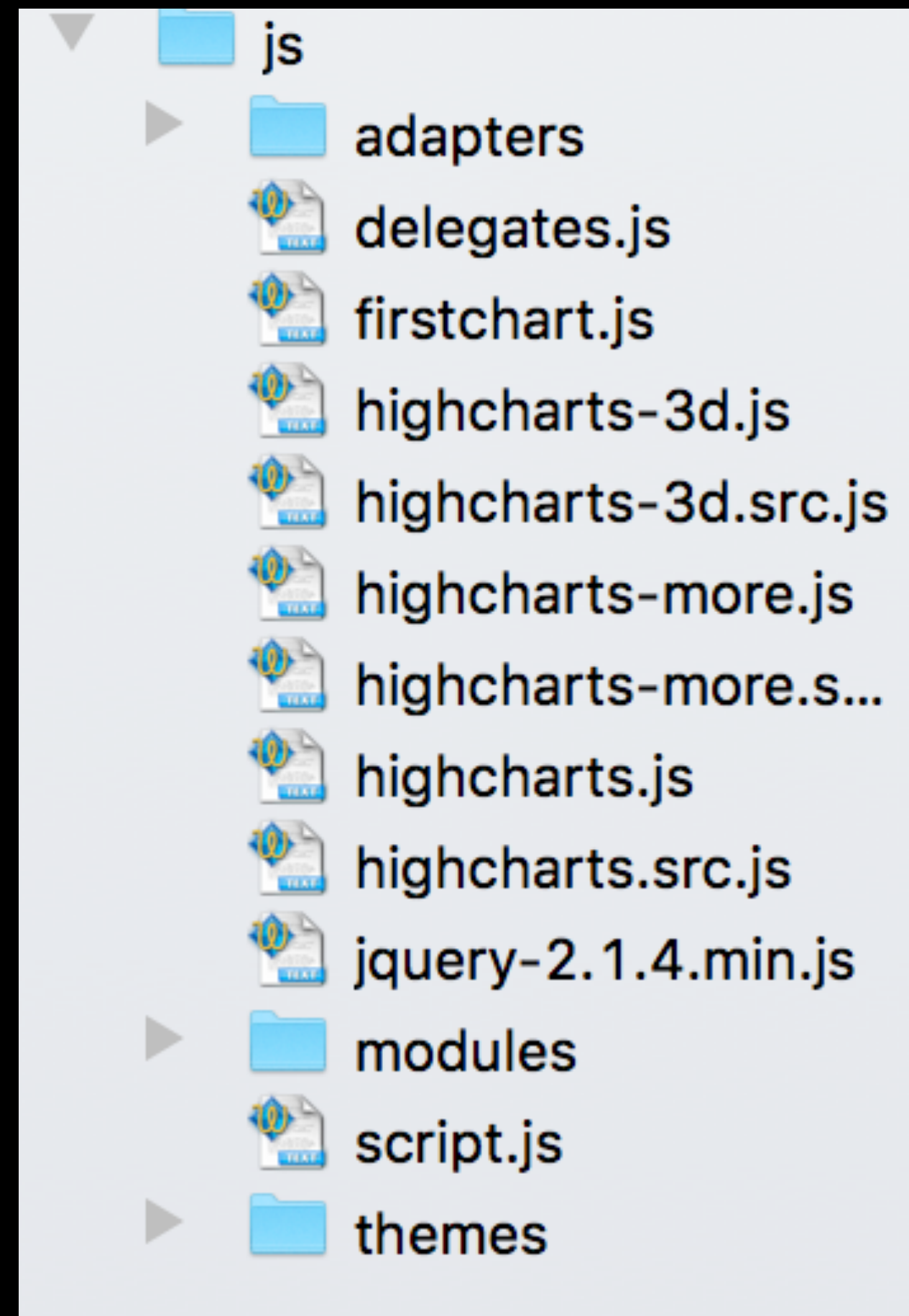
To download jQuery, go to

<http://jquery.com/download/>

(You can also find this link on our syllabus site. Use v2 for our class.)

LOADING THE NECESSARY JAVASCRIPT FILES

After you have downloaded each Javascript file, add it to your **js** folder in your website folder.



LOADING THE NECESSARY JAVASCRIPT FILES

Then, add links to these files in your <head> tag of the HTML page where you want to use Highcharts. Two things to remember:

1. Put your Javascript script tags after your CSS stylesheets.
2. Make sure jQuery precedes the Highcharts!

```
<script src="js/jquery-2.1.4.min.js"></script>  
<script src="js/highcharts.js"></script>
```


FIRST CHART

Your HTML document should have a div where you want the chart to appear. Give it an ID, and set up its width and height as part of a style attribute.

```
<div id="delegates" style="width:100%; height:450px;"></div>
```

FIRST CHART

Your HTML document also needs to call in the chart you will be making. Add this script tag to the bottom of your page, just before the closing body tag. (It can actually appear anywhere, but this is conventional.)

Now, create an empty text document called “delegates.js” inside your JS folder.

```
<div id="delegates" style="width: 100%; height:450px;"></div>
```

```
<script src="js/delegates.js"></script>
```

FIRST CHART

Begin by using the jQuery "\$" identifier to instantiate the chart.

This is a built-in jQuery function, and is shorthand for launching the chart when the page loads (or in other words, when the document is ready).

```
$ ( ) ;
```

is the same as

```
$( document ). ready ( ) ;
```

FIRST CHART

Of course, you want this Javascript to actually do something, so you need to create a function that will run once the page has loaded. Add the function, and a set of curly braces to contain your programming.

```
$(function() {  
  
});
```


FIRST CHART

Next, you will add the `construct` method for the chart. Inside the parentheses after the identifier, you will add, as a string value, the chart's id, based on your HTML page.

```
$(function() {  
    $('#delegates').highcharts({  
    });  
});
```

FIRST CHART

There are a number of parts, or “concepts,” to Highcharts charts that you next need to establish through the code. The first is to create the chart itself, then to identify the title, axes and data, which is compiled within the “series” array.

```
$(function () {  
    $('#delegates').highcharts({  
        chart: {},  
        title: {},  
        xAxis: {},  
        yAxis: {},  
        series: []  
    });  
});
```

FIRST CHART

First, you declare the chart type inside the block for "chart." If this is the only line in your code block for "chart," you won't need a comma at the end, but if you have more than one line, use a comma at the end.

```
$(function () {  
    $('#delegates').highcharts({  
        chart: {  
            type: 'bar'  
        },  
        title: {},  
        xAxis: {},  
        yAxis: {},  
        series: []  
    });  
});
```

FIRST CHART

Second, add the title text. This is a string value set to the "text" object within the title object.

```
$(function () {
    $('#delegates').highcharts({
        chart: {
            type: 'bar'
        },
        title: {
            text: 'GOP delegate count'
        },
        xAxis: {},
        yAxis: {},
        series: []
    });
});
```


FIRST CHART

The X axis is where you add the categories for your bar chart. These are expressed as an array – a series of string values, in order, separated by commas.

```
$(function () {  
    $('#delegates').highcharts({  
        chart: {  
            type: 'bar'  
        },  
        title: {  
            text: 'GOP delegate count'  
        },  
        xAxis: {  
            categories: ['Trump', 'Cruz', '  
        },  
        yAxis: {},  
        series: []  
    });  
});
```

FIRST CHART

The Y axis on a bar chart displays horizontally (vertically for a column chart). You can control whether the axis title displays or not using options, but it is on by default; if you leave out the Y axis, it still titles the axis with the word “Values.” We will add a line of text explaining what the numbers in the chart are.

```
$(function () {
    $('#delegates').highcharts({
        chart: {
            type: 'bar'
        },
        title: {
            text: 'GOP delegate count'
        },
        xAxis: {
            categories: ['Trump', 'Cruz', '']
        },
        yAxis: {
            title: {
                text: 'Through Monday'
            }
        },
        series: []
    });
});
```

FIRST CHART

Finally — at least in terms of creating our first chart — we have the series object. This is where all the data for the chart lives. It is placed within an array — that is, within a set of square brackets, similar to how we constructed the categories. The order is important: the numbers will pair to the strings in the categories array in the Xaxis object.

```
chart: {
  type: 'bar'
},
title: {
  text: 'GOP delegate count'
},
xAxis: {
  categories: ['Trump', 'Cruz', '']
},
yAxis: {
  title: {
    text: 'Through Monday'
  }
},
series: []
});
});
```

FIRST CHART

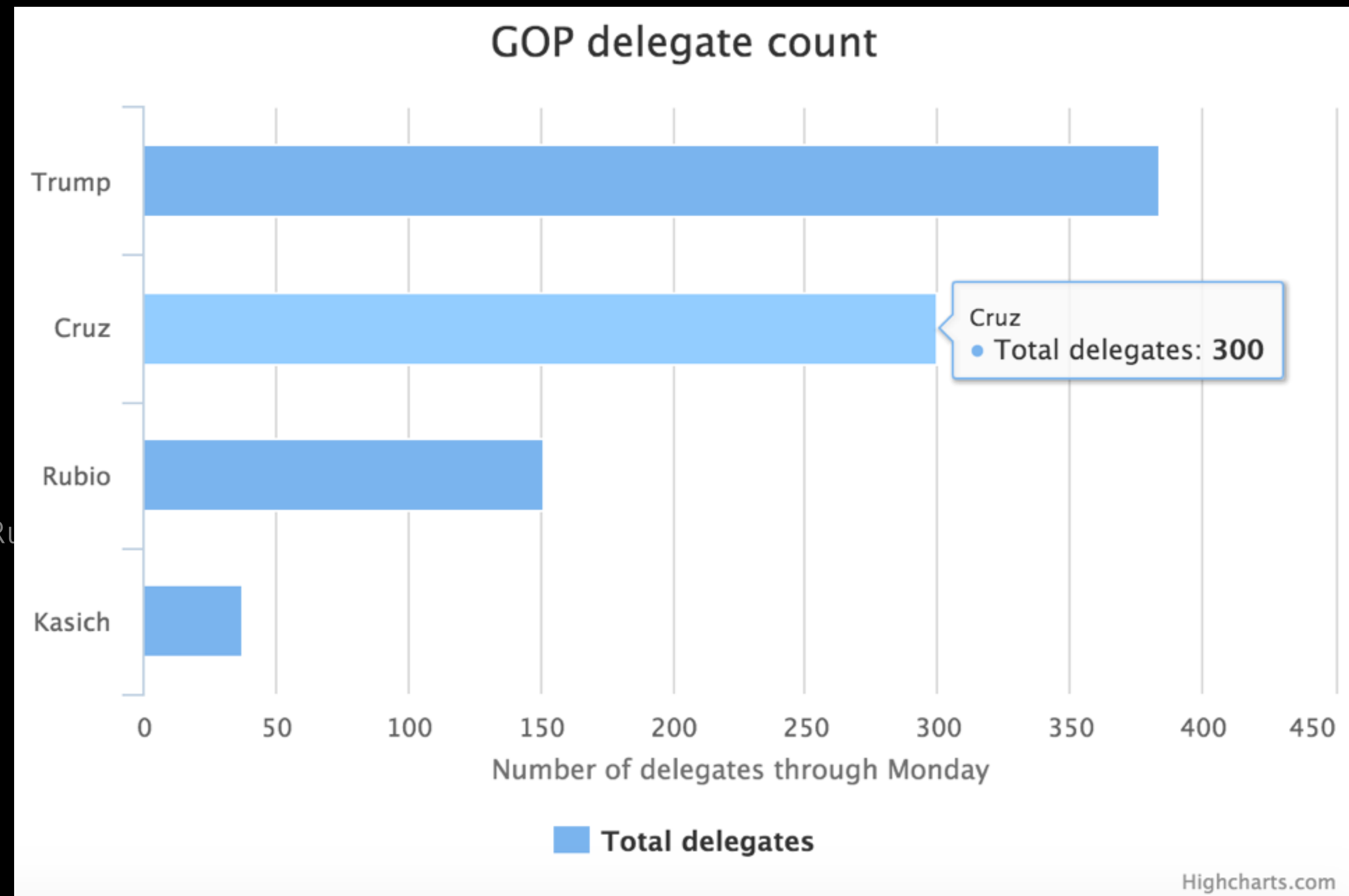
The series includes another array: the data itself, based on the data object. This must be contained within a set of curly braces. There also is a name value in the basic construction. This will, by default, appear as text within the tooltip, which the user will see when hovering over the bar itself.

```
title: {
  text: 'GOP delegate count'
},
xAxis: {
  categories: ['Trump', 'Cruz', '']
},
yAxis: {
  title: {
    text: 'Through Monday'
  }
},
series: [{
  name: 'Delegates won',
  data: [384, 300, 151, 37]
}]
});
});
```


FIRST CHART

Your first chart should be ready to go. Preview in the browser.

```
$(function () {
  $('#delegates').highcharts({
    chart: {
      type: 'bar'
    },
    title: {
      text: 'GOP delegate count'
    },
    xAxis: {
      categories: ['Trump', 'Cruz', 'Rubio', 'Kasich']
    },
    yAxis: {
      title: {
        text: 'Through Monday'
      }
    },
    series: [{
      name: 'Delegates won',
      data: [384, 300, 151, 37]
    }]
  });
});
```



SETTING OPTIONS

Naturally, we want to be able to extend our efforts beyond this. Highcharts offers many, many options for customizing your charts, similar to Google Charts.

```
$(function () {
  $('#delegates').highcharts({
    chart: {
      type: 'bar'
    },
    title: {
      text: 'GOP delegate count'
    },
    xAxis: {
      categories: ['Trump', 'Cruz', '']
    },
    yAxis: {
      title: {
        text: 'Through Monday'
```

SETTING OPTIONS

You can set certain customizations globally, while others you will want to set more specifically. For example, we can change the typography of the entire chart from the default to the type you are using on your site. To do this, add a style object to the chart object, and within the curly braces, set the fontFamily (notice the camel case) to your font of choice.

```
$(function () {  
  $('#delegates').highcharts({  
    chart: {  
      type: 'bar',  
      style: {  
        fontFamily: 'roboto'  
      }  
    },  
    title: {  
      text: 'GOP delegate count'  
    },  
    xAxis: {  
      categories: ['Trump', 'Cruz', '']  
    }  
  })  
})
```

SETTING OPTIONS

Within the chart object, you can also set values for the padding in the chart, using the "spacing" term, as well as its background color (in camelCase, of course).

Note that these are outside the style object.

```
$(function () {
  $('#delegates').highcharts({
    chart: {
      type: 'bar',
      style: {
        fontFamily: 'roboto'
      },
      spacing: 30,
      backgroundColor: 'rgba(255,255,255,0.2)',
    },
    title: {
      text: 'GOP delegate count'
    },
    xAxis: {

```


SETTING OPTIONS

Try making the title text larger, bolder, and aligned to the left. The first two of these need the "style" object, but the last does not.

```
        backgroundColor: 'rgba(255,255,255,0.5)',
    },
    title: {
      text: 'GOP delegate count',
      style: {
        fontWeight: '900',
        fontSize: '22px'
      },
      align: 'left'
    },
    xAxis: {
      categories: ['Trump', 'Cruz', 'Clinton']
    },
    yAxis: {
      title: {
        text: 'Through Monday'
      }
    },
    series: [{
      name: 'Delegates won'
    }, {
      name: 'Delegates needed'
    }
  ]
}
```

SETTING OPTIONS

Optionally, you can add a subtitle beneath the title. It has the same construction as the title object, so copy and paste that chunk of your script and change the values.

```
        backgroundColor: 'rgba(255,255,255,0.1)',
    },
    title: {
        text: 'GOP delegate count',
        style: {
            fontWeight: '900',
            fontSize: '22px'
        },
        align: 'left'
    },
    subtitle : {
        text: 'Includes delegates awarded
Rico primary',
        align: 'left',
        style: {
            fontStyle: 'italic'
        }
    },
    xAxis: {
```

SETTING OPTIONS

You can also choose to override certain defaults – for example, to not display certain lines of text in the graphic. In our graphic, we don't really need the legend, which appears in the bottom of the chart and only has one term. Is there any confusion as to what the chart is showing? No. So let's set the legend object an enabled value of "false."

```
$(function () {
  $('#delegates').highcharts({
    legend: {
      enabled: false
    },
    chart: {
      type: 'bar',
      style: {
        fontFamily: 'roboto'
      },
      spacing: '30',
      backgroundColor: 'rgba(255,255,255,0.5)',
    },
    title: {
      text: 'GOP delegate count',
    }
  });
});
```

SETTING OPTIONS

Highcharts charts come with their own credit line, but we can suppress that the same way.

```
$(function () {  
  $('#delegates').highcharts({  
    legend: {  
      enabled: false  
    },  
    credits: {  
      enabled: false  
    },  
    chart: {  
      type: 'bar',  
      style: {  
        fontFamily: 'roboto'  
      },  
      spacing: '30',  
      backgroundColor: 'rgba(255,255,2
```

SETTING OPTIONS

Alternately, you can change the text so it is your credit, and not Highcharts. By default, the credit line link to highcharts.com, but you can change this, too.

```
$(function () {
  $('#delegates').highcharts({
    legend: {
      enabled: false
    },
    credits: {
      text: 'YourName / IDV460',
      href: 'index.html'
    },
    chart: {
      type: 'bar',
      style: {
        fontFamily: 'roboto'
      },
      spacing: '30',
      legend: {
        x: 10,
        y: 10,
        width: 100,
        height: 100
      }
    }
  });
});
```

SETTING OPTIONS

To change the color of the bars, you can add a "Color" rule to the series. If you were creating a chart based on multiple values, you would change each bar color within each of the values within the series object.

```
        fontStyle: 'italic'
      }
    },
    xAxis: {
      categories: ['Trump', 'Cruz', '']
    },
    yAxis: {
      title: {
        text: 'Through Monday'
      }
    },
    series: [{
      name: 'Delegates won',
      data: [384, 300, 151, 37],
      color: 'red'
    }]
  });
});
```


SETTING OPTIONS

You can also choose to label the bars themselves on the chart. If there is a compelling reason to not show them — like the bars are too small or too narrow — they should remain hidden, but in our chart, we can probably add them in. This means overriding the default value for the `dataLabels` object, which is to hide them (or `enabled: 'false'`).

```
    xAxis: {
      categories: ['Trump', 'Cruz', '']
    },
    yAxis: {
      title: {
        text: 'Through Monday'
      }
    },
    series: [{
      name: 'Delegates won',
      data: [384, 300, 151, 37]
      color: 'red',
      dataLabels: {
        enabled: 'true'
      }
    }]
  });
});
```

SETTING OPTIONS

Modify the look of these labels with a few more lines of code within the dataLabels object. Change the color and position (default is black and left, respectively) to white and right, and add a slight shift in the x position so they are inside the bars.

```
xAxis: {
  categories: ['Trump', 'Cruz', '']
},
yAxis: {
  title: {
    text: 'Through Monday'
  }
},
series: [{
  name: 'Delegates won',
  data: [384, 300, 151, 37]
  color: 'red',
  dataLabels: {
    enabled: 'true',
    color: 'white',
    align: 'right',
    x: -10
  }
}]
```