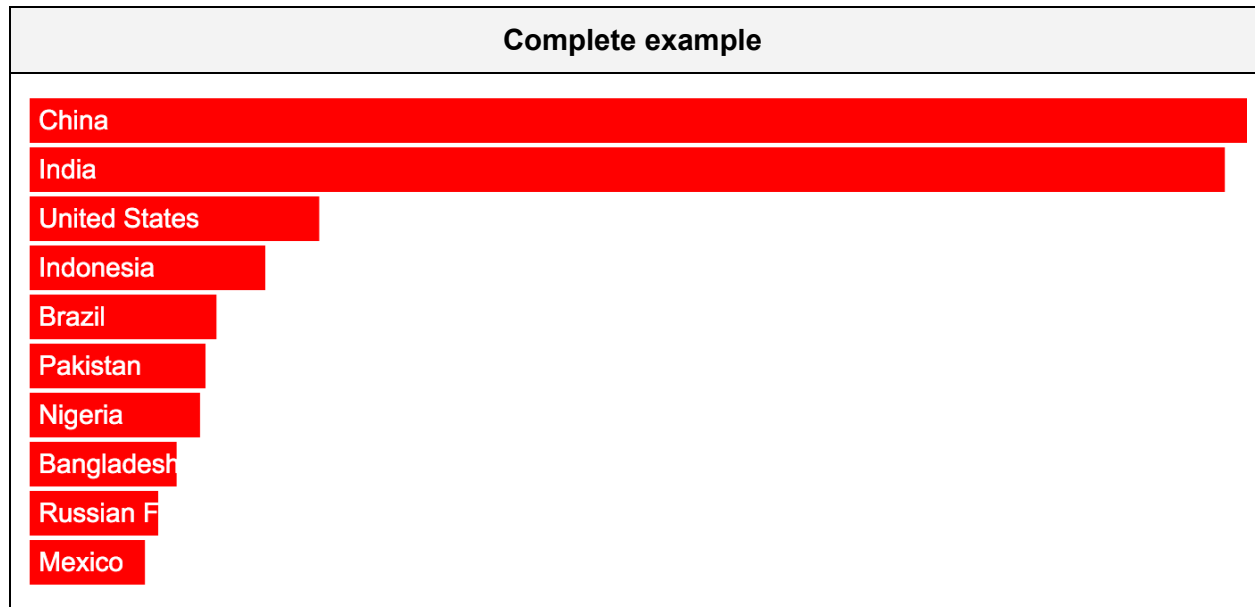


Top 10 Countries

Introduction

This tutorial will teach you the basics of graphing data in a file using HTML `<canvas>`.



Steps

Step 1 - Create your project folder

Use Finder or File Explorer to navigate to your workspace. Create a new folder called **top10**.

CS Tip: files and folders should be 8 characters or less with no spaces or special characters

Step 2 - Create your HTML file

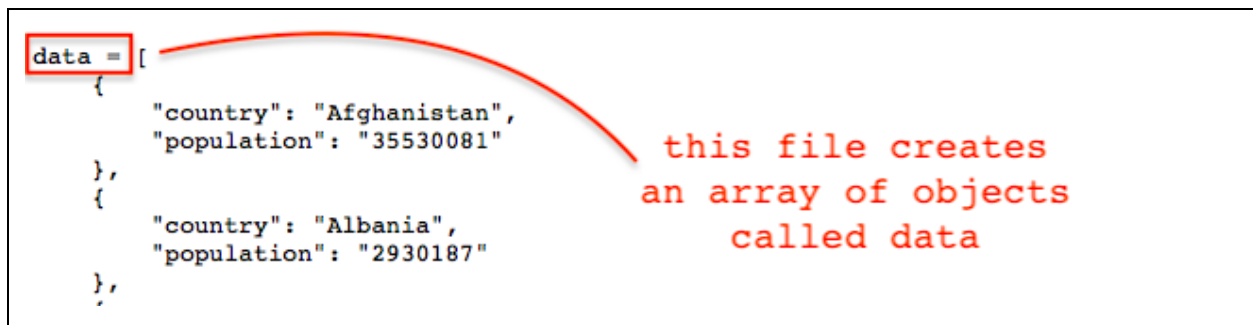
Create a new **top10.html** in your new folder using your HTML editor (i.e., Brackets).



Step 3 - Download the countries.js data file

Download <https://raw.githubusercontent.com/mdjhoel/datacourse/master/countries.js> to your new project folder. This file contains an array of objects called **data**.

CS Tip: keep your data in external files and source it to avoid making your HTML files confusing



Step 4 - Create your canvas with large width and height

Use your HTML editor to create the following code in top10.html.

```
<html>  
  <canvas id="mycanvas" width="1200" height="800"></canvas>  
</html>
```

Step 5 - Source your JavaScript data file in your HTML

This step allows you to access your data, while keeping your HTML small and readable.

```
<html>  
  <canvas id="mycanvas" width="1200" height="800"></canvas>  
  <script src="countries.js"></script>  
</html>
```

Step 6 - Create a <script> area and set up canvas drawing environment

Add the following code to gain control of the <canvas> tag in JavaScript, and set up the '2d' drawing context.

The document.getElementById() function is one of the most commonly used commands in JavaScript - https://www.w3schools.com/jsref/met_document_getelementbyid.asp.

The '2d' drawing context allows access to the drawing tools of <canvas>.

```

<html>
  <canvas id="mycanvas" width="1200" height="800"></canvas>
  <script src="countries.js"></script>
  <script>
    var c = document.getElementById("mycanvas");
    var ctx = c.getContext('2d');
  </script>
</html>

```

Step 7 - Sort and reverse the data array

This may be the most complicated part of this tutorial.

Sorting an array of objects requires you to create a custom function with the built in **.sort** function. The function cycles through all objects in the array and compares a specific object property against each other and sorts it..

In this case we are comparing the object property population. After sorting on population, we reverse the data using the built in array function **.reverse** to get the data in descending order.

```

<script>
  var c = document.getElementById("mycanvas");
  var ctx = c.getContext('2d');

  data.sort(function(a, b){
    return a.population - b.population;
  })
  data.reverse();
</script>

```

Step 8 - Draw first data object as a red rectangle

The **ctx** variable represents the 2d drawing toolkit of the <canvas> tag.

To change the drawing colour use the **fillStyle** command.

To draw a rectangle use the **fillRect(x,y,width,height)** function.

See all <canvas> drawing commands at: https://www.w3schools.com/tags/ref_canvas.asp.

Notice that the rectangle is drawn from the top left of the canvas (0,0) to the width of first data object in the array (0 or first element, or country: China). The value is of China's population is so large it must be divided by 1000000 so it will fit within the <canvas> width.

```

<script>
  var c = document.getElementById("mycanvas");
  var ctx = c.getContext('2d');

  data.sort(function(a, b){
    return a.population - b.population;
  })
  data.reverse();

  ctx.fillStyle = "red";
  ctx.fillRect(0,0,data[0].population/1000000,50);

</script>

```

Step 9 - Draw the country name on top of the country rectangle

Commands run sequentially in a computer program. Therefore, to draw a country name on top of the rectangle, you must ...

1. Specify a contrasting colour for text using `fillStyle`
2. Set the font to an appropriate size using `font`
3. Draw the text in the correct x,y location within the rectangle using `fillText`



China

```

<script>
  var c = document.getElementById("mycanvas");
  var ctx = c.getContext('2d');

  data.sort(function(a, b){
    return a.population - b.population;
  })
  data.reverse();

  ctx.fillStyle = "red";
  ctx.fillRect(0,0,data[0].population/1000000,50);
  ctx.fillStyle = "white";
  ctx.font = "30px Arial";
  ctx.fillText(data[0].country,10,35);

</script>

```

Step 10 - Create algorithm to cycle through top 10 countries

An algorithm is a step by step process to solve a problem. In this step, you will add code to create an algorithm to draw the top 10 country bars to complete the graph.

Given that all bars will start at x value of 0 and each bar will be 50 pixels in height, here are the steps to loop through the top 10 countries and draw them on your web page.

1. Create a variable **y** and set it to 0
2. Create a **for loop** that creates a counter variable called **i** and loops to 10
3. Swap out hard coded y values and replace with y variable
4. Swap out hard coded 0 value (first element - China) for counter variable i
5. Add **35 pixels to the y** value to ensure that the text is drawn inside the rectangle (this can be changed depending on the size and font specified)
6. **Increment y value** by 55 pixels each loop, so the next rectangle will start below current

```
<script>
  var c = document.getElementById("mycanvas");
  var ctx = c.getContext('2d');

  data.sort(function(a, b){
    return a.population - b.population;
  })
  data.reverse();

  var y = 0
  for (i = 0; i<10; i++) {
    ctx.fillStyle = "red";
    ctx.fillRect(0,y,data[i].population/1000000,50);
    ctx.fillStyle = "white";
    ctx.font = "30px Arial";
    ctx.fillText(data[i].country,10,y+35);
    y = y + 55;
  }
</script>
```

China

India

United States

Ta Daaaa! Voila!

