

Exercise 14: Libraries

For this exercise, you'll use a JavaScript module (that is, library) to draw a chart.

Obtain the module

We are going to be using an open source charting module called AwesomeChart. It is hosted on GitHub.

Note: Because this project is open source, it could change at any time. If this exercise does not work, please contact me at once so I can update it.

Follow these steps to obtain the module:

1. Navigate to the project URL: <https://github.com/cyberpython/AwesomeChartJS>
2. Click on the **Clone or Download** button and choose **Download Zip**
3. Unzip the downloaded file.
4. Copy the file **awesomechart.js** into the folder where you are going to do this exercise.

Starting Code

Create a new file in your text editor and paste in this code, which is the basics to get started.

```
<!DOCTYPE HTML>
<html>
<body>

<script

</script>

</body>
</html>
```

Save this as **module.html**.

Import the library

Before the **<script>** tag, another **<script>** tag that imports the module. Use this line, which will point to the JavaScript file you just downloaded.

```
<script type="application/javascript" src="awesomechart.js"></script>
```

Add a canvas and code

Before the **<script>** tags, add a **canvas** UI element. A canvas is used for placing graphics on the screen. The text inside the canvas element will display if your browser doesn't support HTML5, which is required for this to work.

```
<canvas id="canvas1" width="300" height="300">
  Your web-browser does not support the HTML 5 canvas element.
```

</canvas>

Now add a function named **drawChart** that is called when the page loads using the **onload** attribute of the **<body>** tag. Modify the **<body>** tag to be this:

```
<body onload="drawChart()">
```

Finally, add the **drawChart** function into the **<script>** element. This code is modified from the **AwesomeChart** sample code. Note that the **!!** is a shortcut that converts a JavaScript object to true if something is not null, and false if it is null.

```
function drawChart(){
  if (!!document.createElement('canvas').getContext){ //check that the canvas
                                                    // element is supported

    var mychart = new AwesomeChart('canvas1');
    mychart.title = "Product Sales - 2010";
    mychart.data = [1532, 3251, 3460, 1180, 6543];
    mychart.labels = ["Desktops", "Laptops", "Netbooks", "Tablets", "Smartphones"];
    mychart.draw();
  }
}
```

Note that the line that creates the **AwesomeChart** object references the ID of the canvas we created earlier (**canvas1**).

Save and open in the browser. You should see a bar chart.



Wasn't that easy? You didn't need to write any code to create charts.

Play around with the **mychart.data** and **mychart.labels** arrays, save and refresh, and watch how your changes affect the chart.

Take a Look at How I've Done It

If it's not working for you, you can look at my versions of the code:

<http://sdkbridge.com/prog1/Exercise14Answers.pdf>.