Fakultas Ilmu Komputer

Learning is the process of acquiring new understanding, knowledge, behaviors, skills, values, attitudes, and preferences.

JavaScript Graphics

Welcome to *the* Front-end Web Development *course*

1 - JavaScript Graphics

Gender - Manufacture Dept.
29 Employees
7
22
Male Female
Gender - Sales Dept.
21 Employees
8
13
Male Female
Gender - Engineering Dept.
43 Employees
9
34
Male Female
Gender - Finance Dept.
7 Employees
1
6
Male Female
Number of Employees by Monthly Salary
> $8000
13
$7000 - $8000
17
30
40
Employees Composition by Grade
100 Employees
3 4 5 6 7 8 9 10 11 12
Number of Employees by Year
30
20
10
0
16
14
2012
2013
Women First Granted Suffrage at National Level (by Year)
Before 1907 1907 - 1920 1920 - 1940 1940 - 1950 1950 - 1960 1960 - 1970 1970 - 1980 After 1980

Web programmers dream to create beautiful graphic designs in their webpage.

Multiple Drill Down in Dashboard Mode

Can JavaScript *do* graphics?

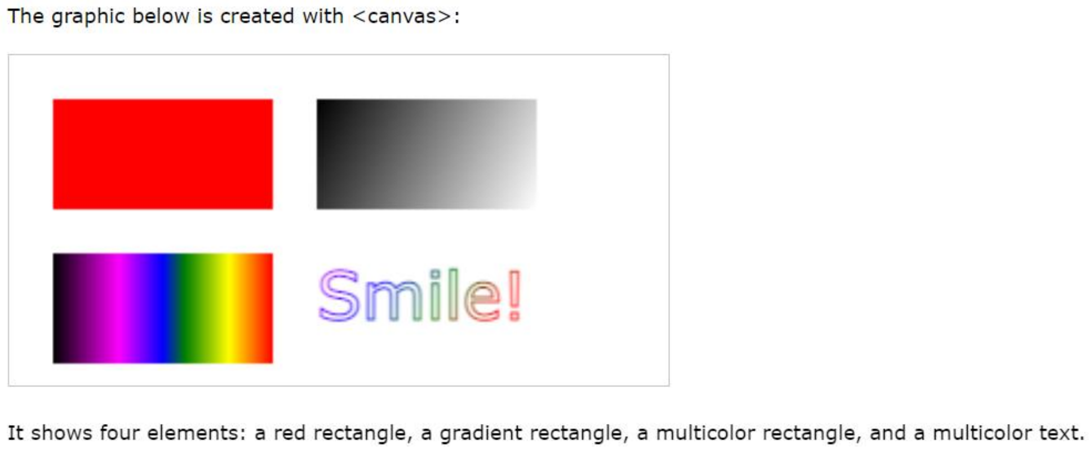
- ✓ Typical uses for graphics include ***banners***, ads from ***sponsoring companies***, and colored ***bullets*** to *highlight important pieces of text*.

- ✓ The JavaScript scripting language can be used to enhance the graphics you put on the Web pages.

- ✓ **JavaScript can be used to <u>dynamically control the graphic content</u> of the page.**

How graphics *are* created in HTML?

The HTML <canvas> element can be used to draw graphics on a web page.



The graphic below is created with <canvas>:

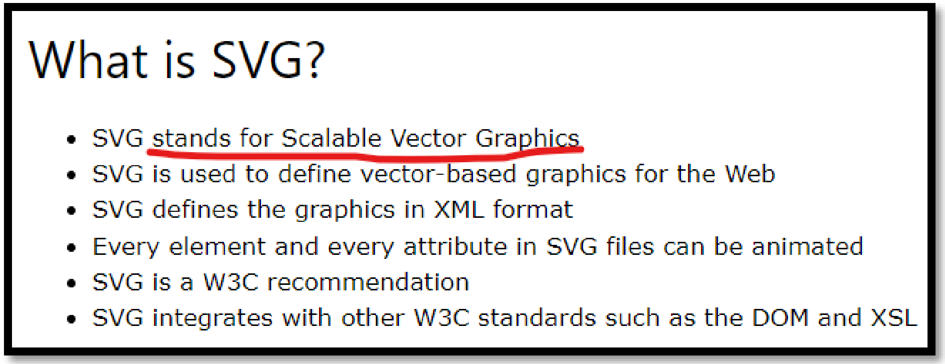
It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

Graphic Libraries

JavaScript libraries to use for both Artificial Intelligence graphs and other charts:

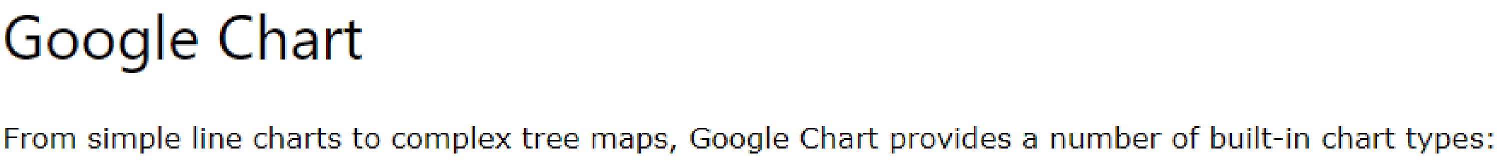
- ✓ Plotly.js
- ✓ Chart.js
- ✓ Google Chart

Plotly.js
Plotly.js is a charting library that comes with over 40 chart types, 3D charts, statistical graphs, and SVG maps.



What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- Every element and every attribute in SVG files can be animated
- SVG is a W3C recommendation
- SVG integrates with other W3C standards such as the DOM and XSL



Google Chart

From simple line charts to complex tree maps, Google Chart provides a number of built-in chart types:

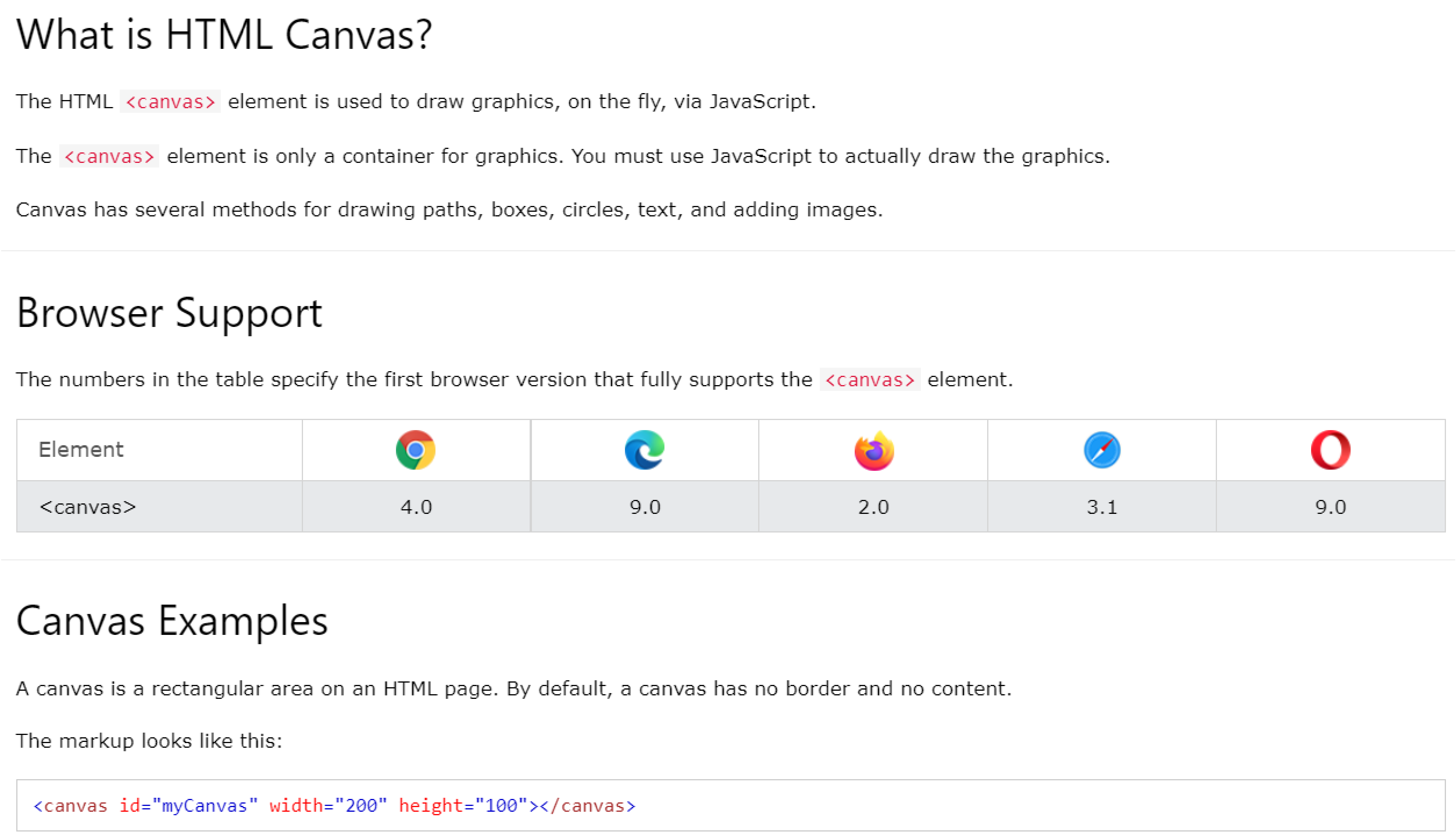
- Scatter Chart
- Line Chart
- Bar / Column Chart
- Area Chart
- Pie Chart
- Donut Chart
- Org Chart
- Map / Geo Chart

Chart.js

Chart.js comes with many built-in chart types:

- Scatter
- Line
- Bar
- Radar
- Pie and Doughnut
- Polar Area
- Bubble

HTML Canvas Graphics



What is HTML Canvas?

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Browser Support

The numbers in the table specify the first browser version that fully supports the `<canvas>` element.

Element					
<canvas>	4.0	9.0	2.0	3.1	9.0

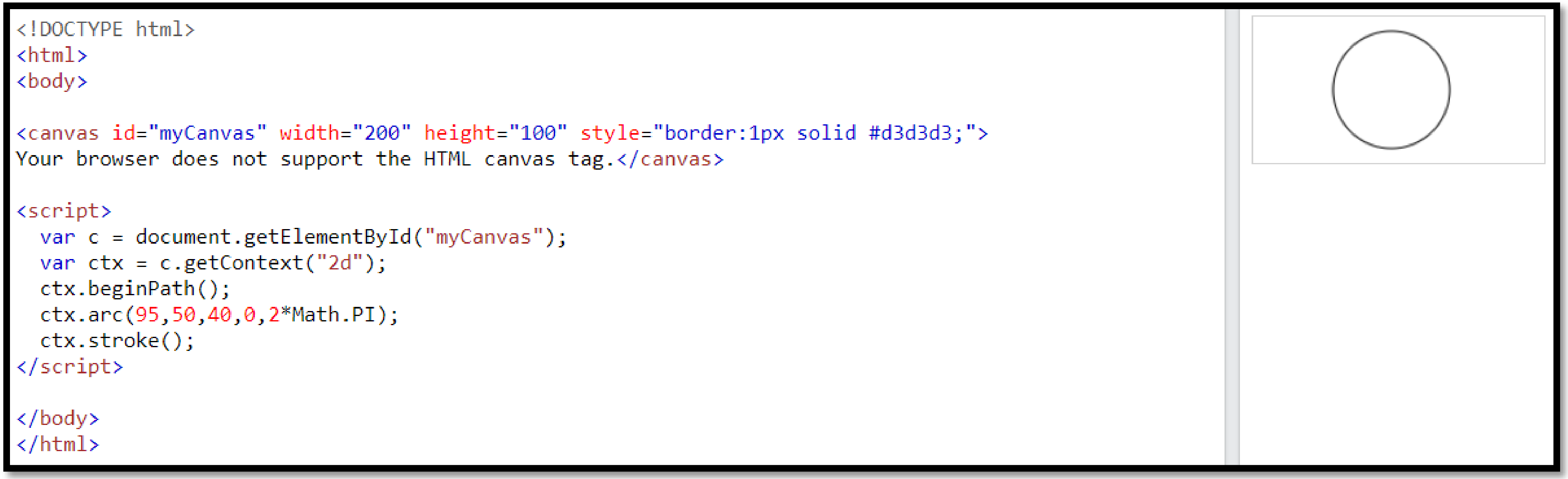
Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Draw a Circle



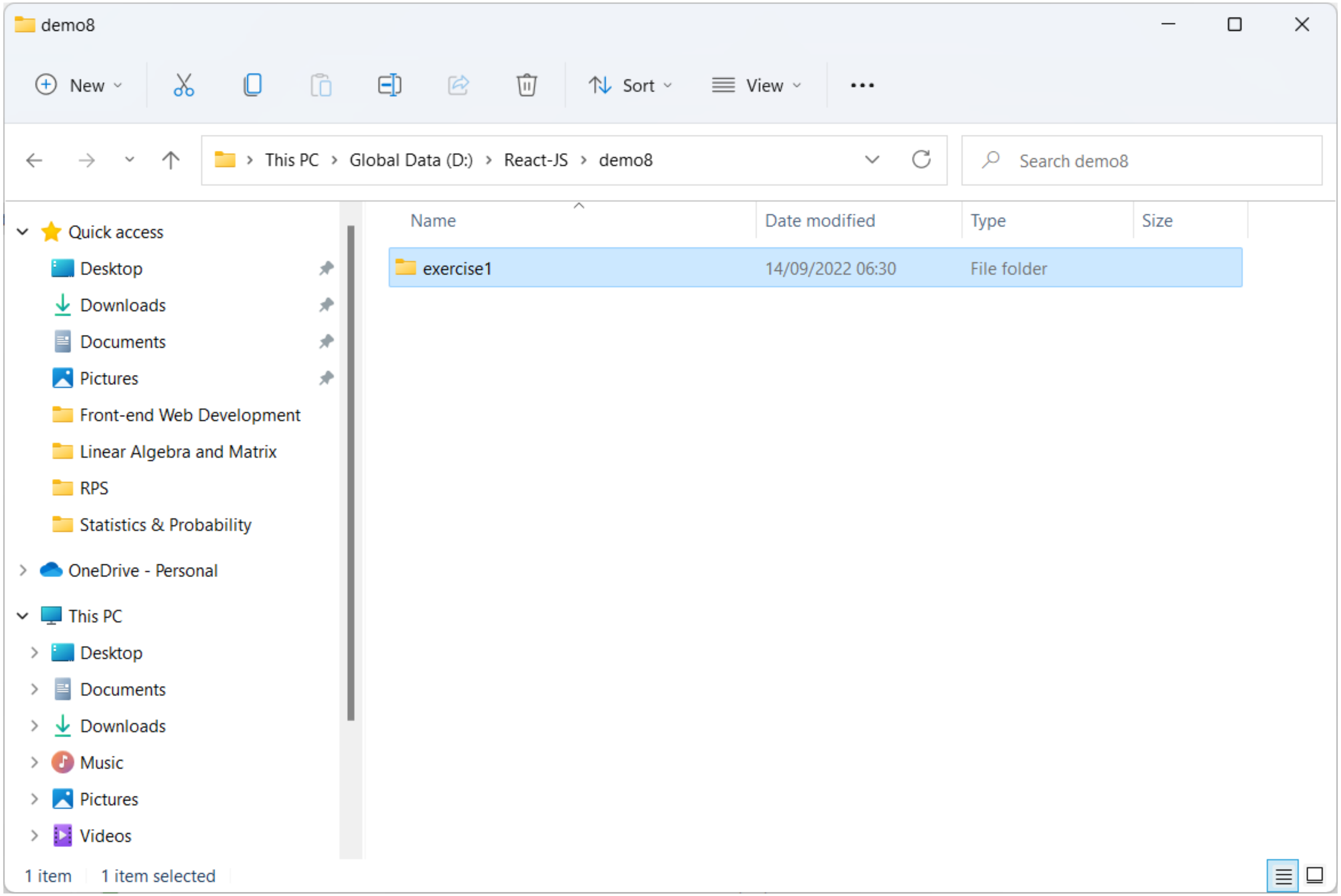
```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML canvas tag.</canvas>

<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");
  ctx.beginPath();
  ctx.arc(95,50,40,0,2*Math.PI);
  ctx.stroke();
</script>

</body>
</html>
```

Canvas has APIs to draw circles and arcs:

arc(x, y, radius, startAngle, endAngle, anticlockwise): Draws an arc that is centered at (x, y) position with radius r starting at startAngle and ending at endAngle, going in the given direction indicated by anticlockwise (*defaulting to clockwise*).

Exercise *for* Students

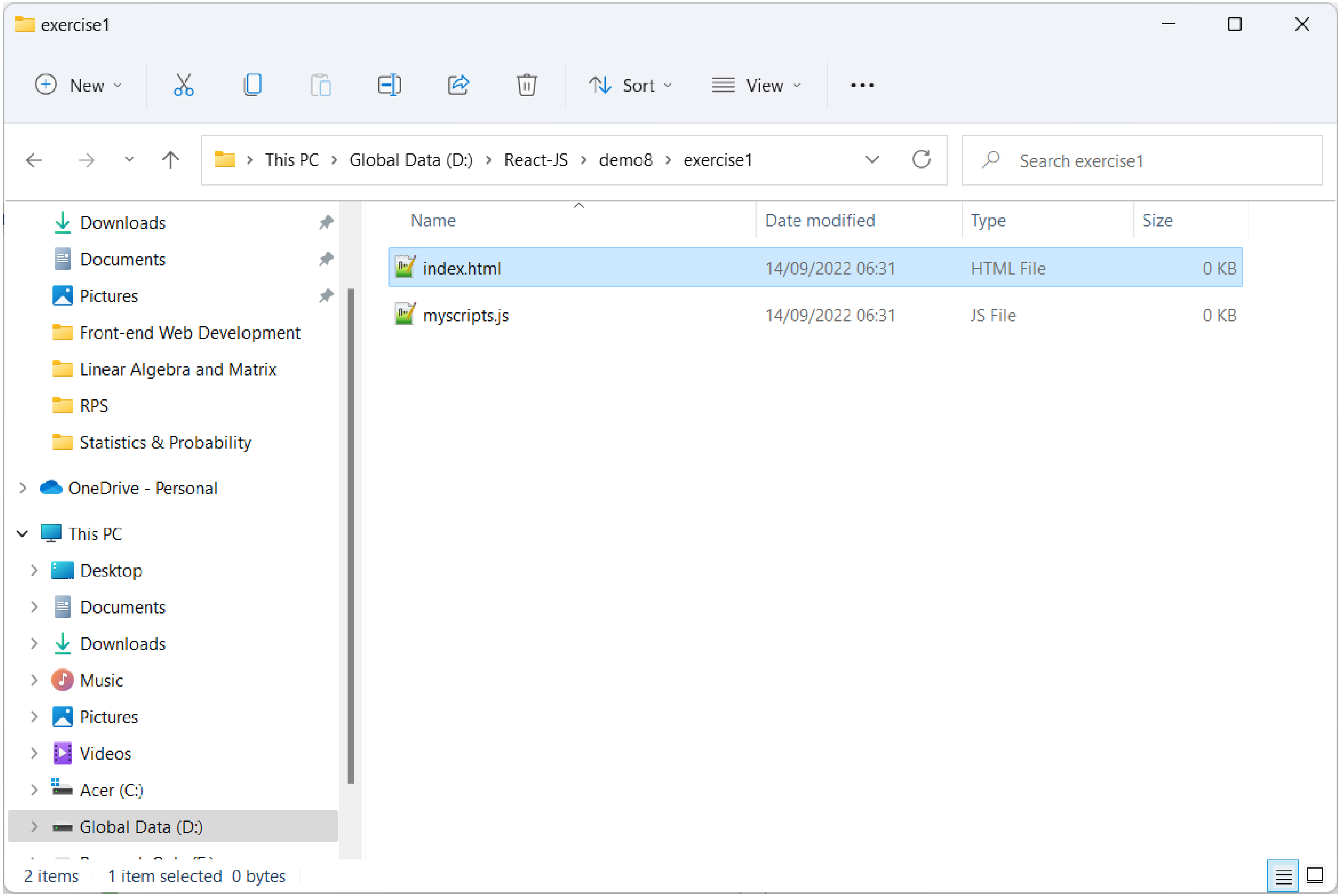
Create New Working Directory



Name	Date modified	Type	Size
exercise1	14/09/2022 06:30	File folder	

1 item 1 item selected

Create 2 files for each exercise



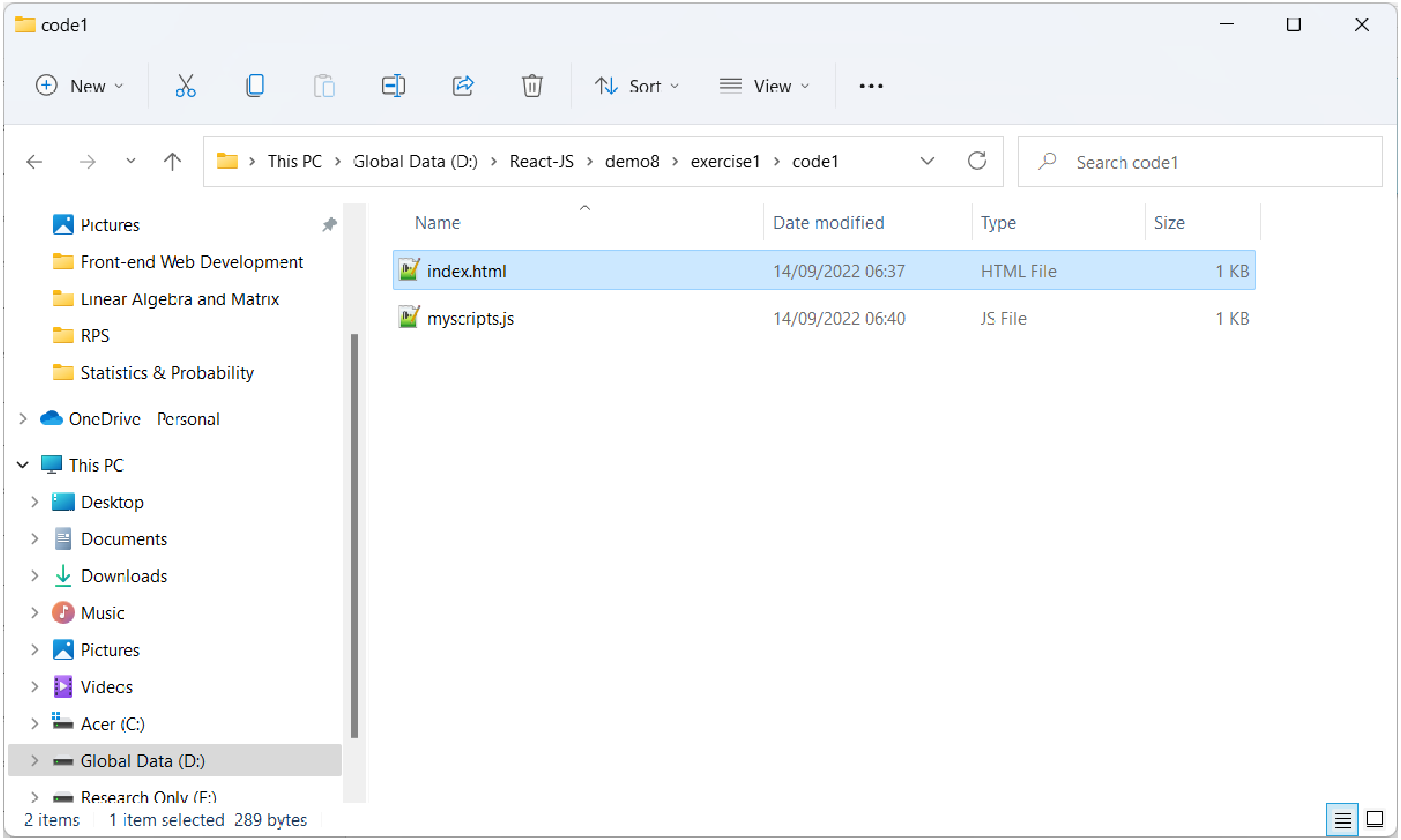
Name	Date modified	Type	Size
index.html	14/09/2022 06:31	HTML File	0 KB
myscripts.js	14/09/2022 06:31	JS File	0 KB

Exercise #1

Plotly.js Javascript Open Source Graphing Library

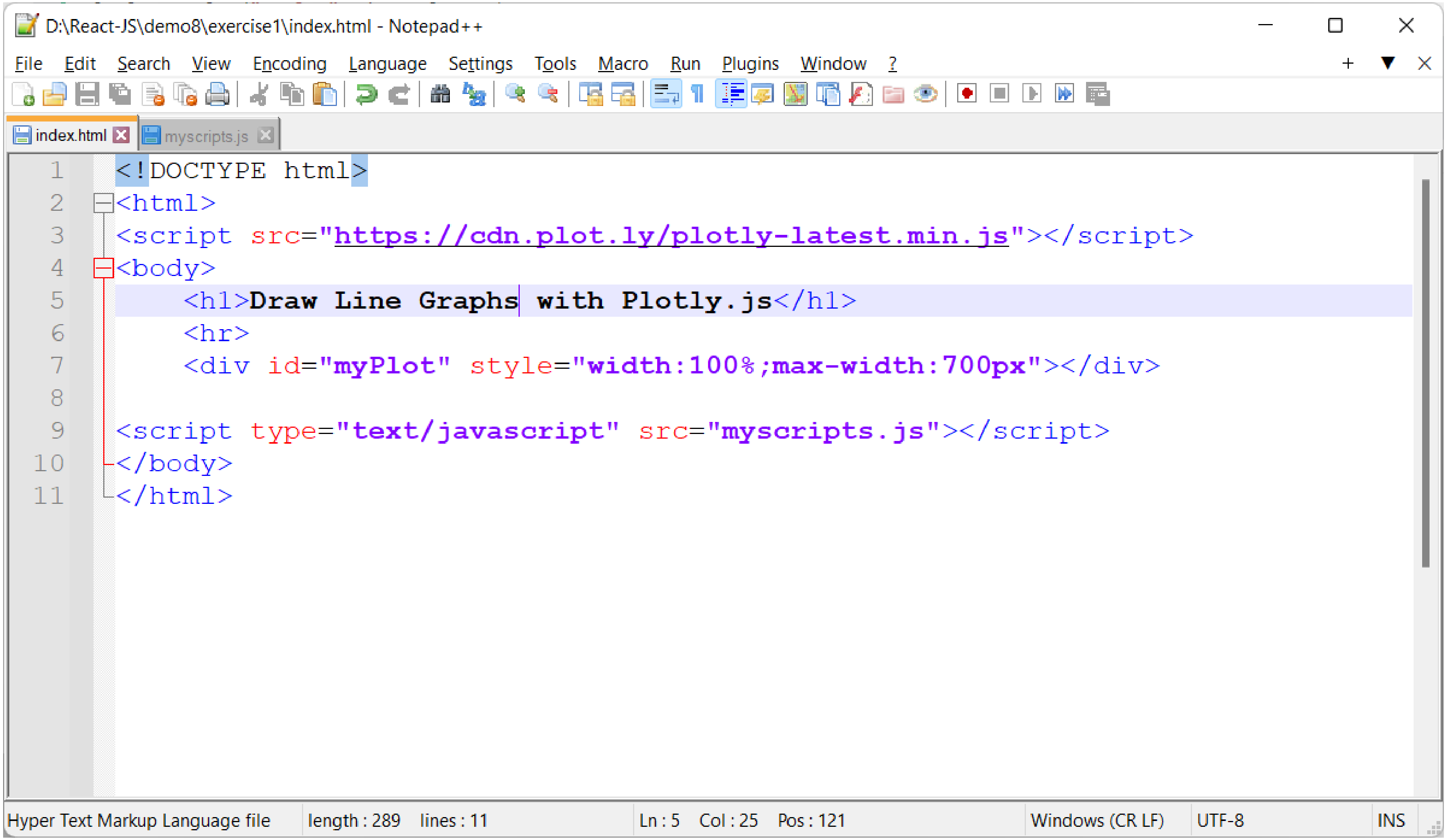
Code #1
(Line Graph)

Create HTML & JavaScript Files



Name	Date modified	Type	Size
index.html	14/09/2022 06:37	HTML File	1 KB
myscripts.js	14/09/2022 06:40	JS File	1 KB

Write Code *in* HTML File

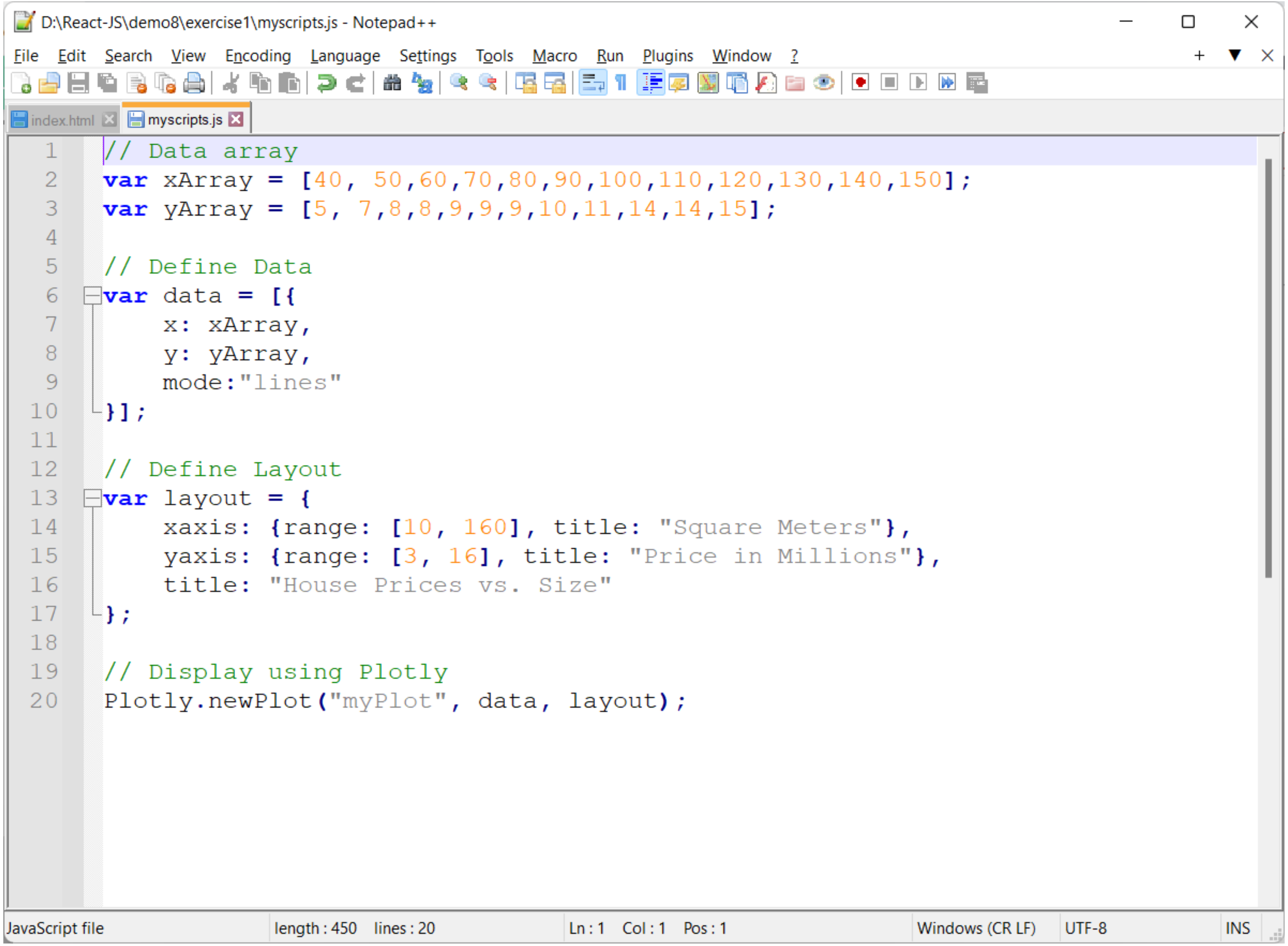


D:\React-JS\demo8\exercise1\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js

```html
<!DOCTYPE html>
<html>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<body>
    <h1>Draw Line Graphs with Plotly.js</h1>
    <hr>
    <div id="myPlot" style="width:100%;max-width:700px"></div>

<script type="text/javascript" src="myscripts.js"></script>
</body>
</html>
```

Hyper Text Markup Language file
length : 289 lines : 11
Ln : 5 Col : 25 Pos : 121
Windows (CR LF)
UTF-8
INS

Write Code *in* Javascript File

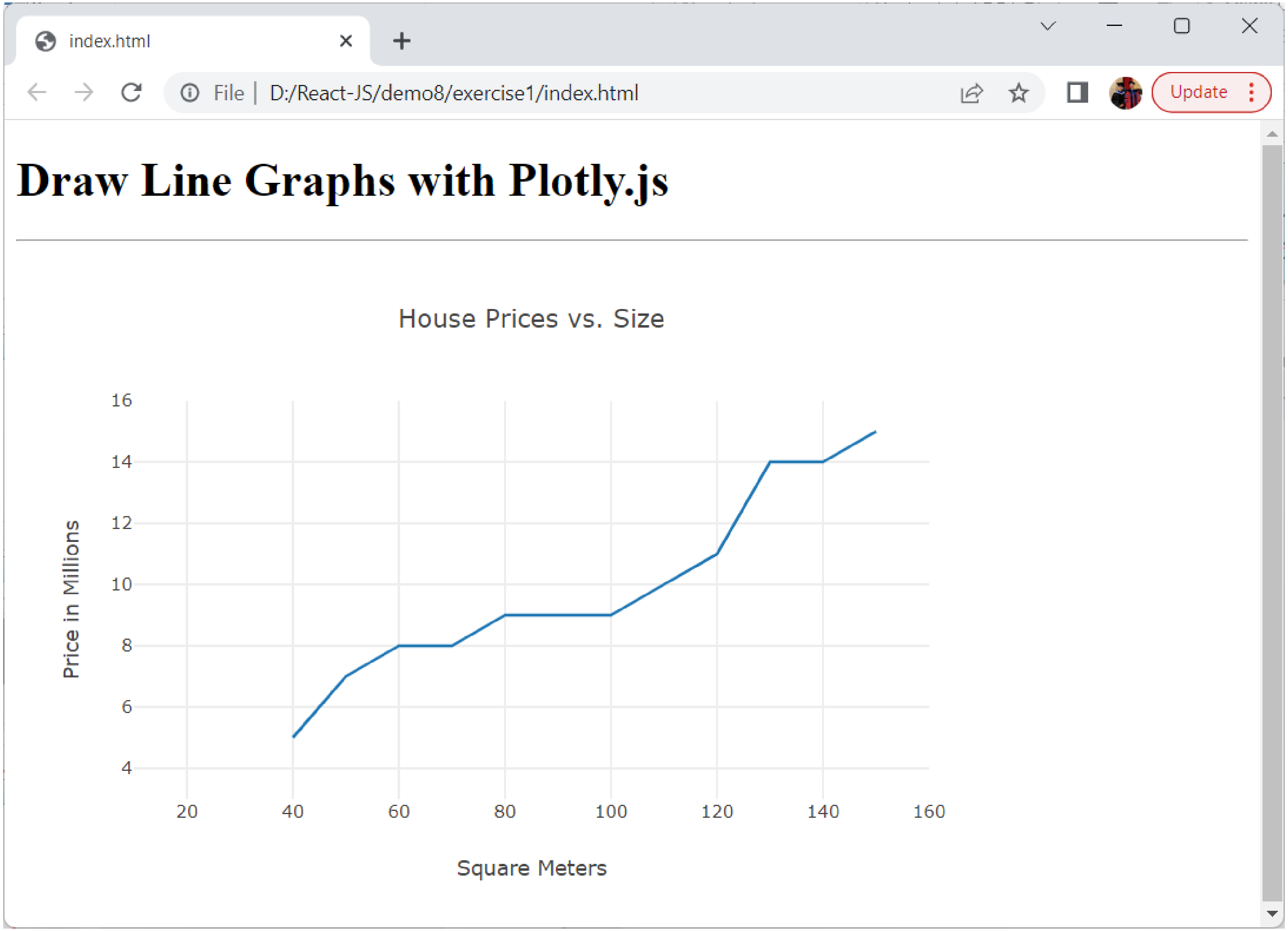


D:\React-JS\demo8\exercise1\myscripts.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js

```
// Data array
var xArray = [40, 50,60,70,80,90,100,110,120,130,140,150];
var yArray = [5, 7,8,8,9,9,9,10,11,14,14,15];

// Define Data
var data = [{
    x: xArray,
    y: yArray,
    mode:"lines"
}];

// Define Layout
var layout = {
    xaxis: {range: [10, 160], title: "Square Meters"},
    yaxis: {range: [3, 16], title: "Price in Millions"},
    title: "House Prices vs. Size"
};

// Display using Plotly
Plotly.newPlot("myPlot", data, layout);
```

JavaScript file
length : 450 lines : 20
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

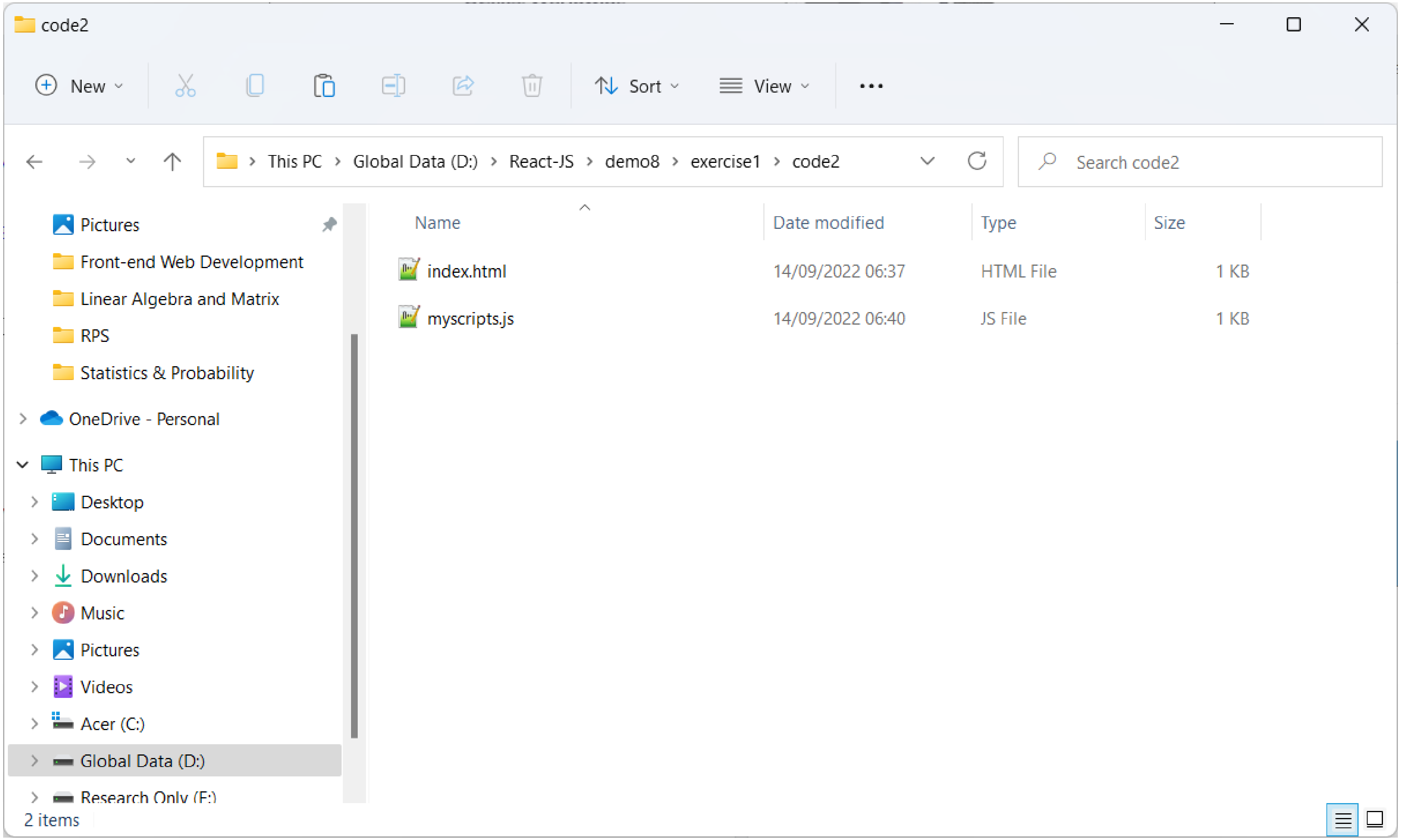
Expected Output



Draw Line Graphs with Plotly.js

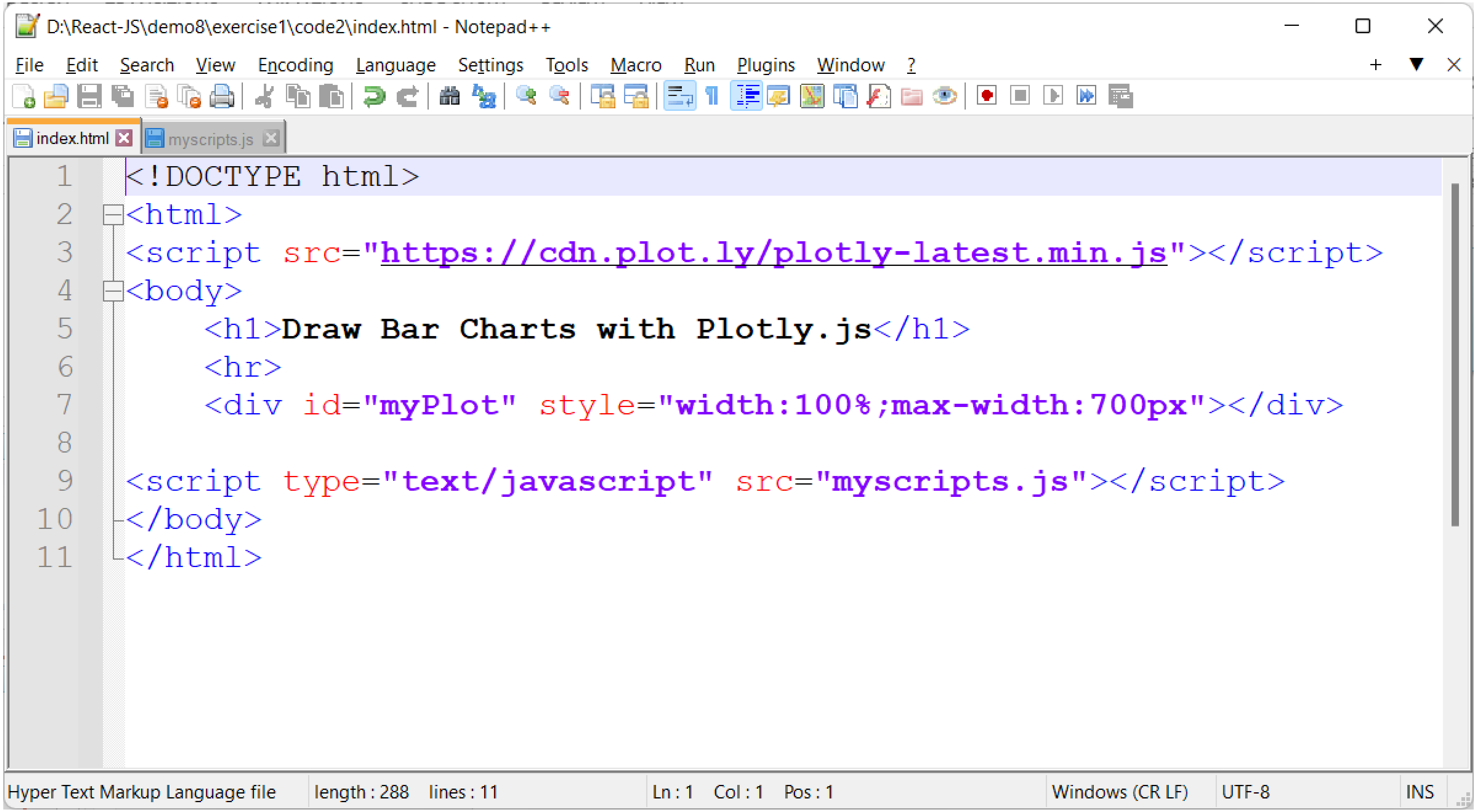
Code #2
(Bar Charts)

Create HTML & JavaScript Files



Name	Date modified	Type	Size
index.html	14/09/2022 06:37	HTML File	1 KB
myscripts.js	14/09/2022 06:40	JS File	1 KB

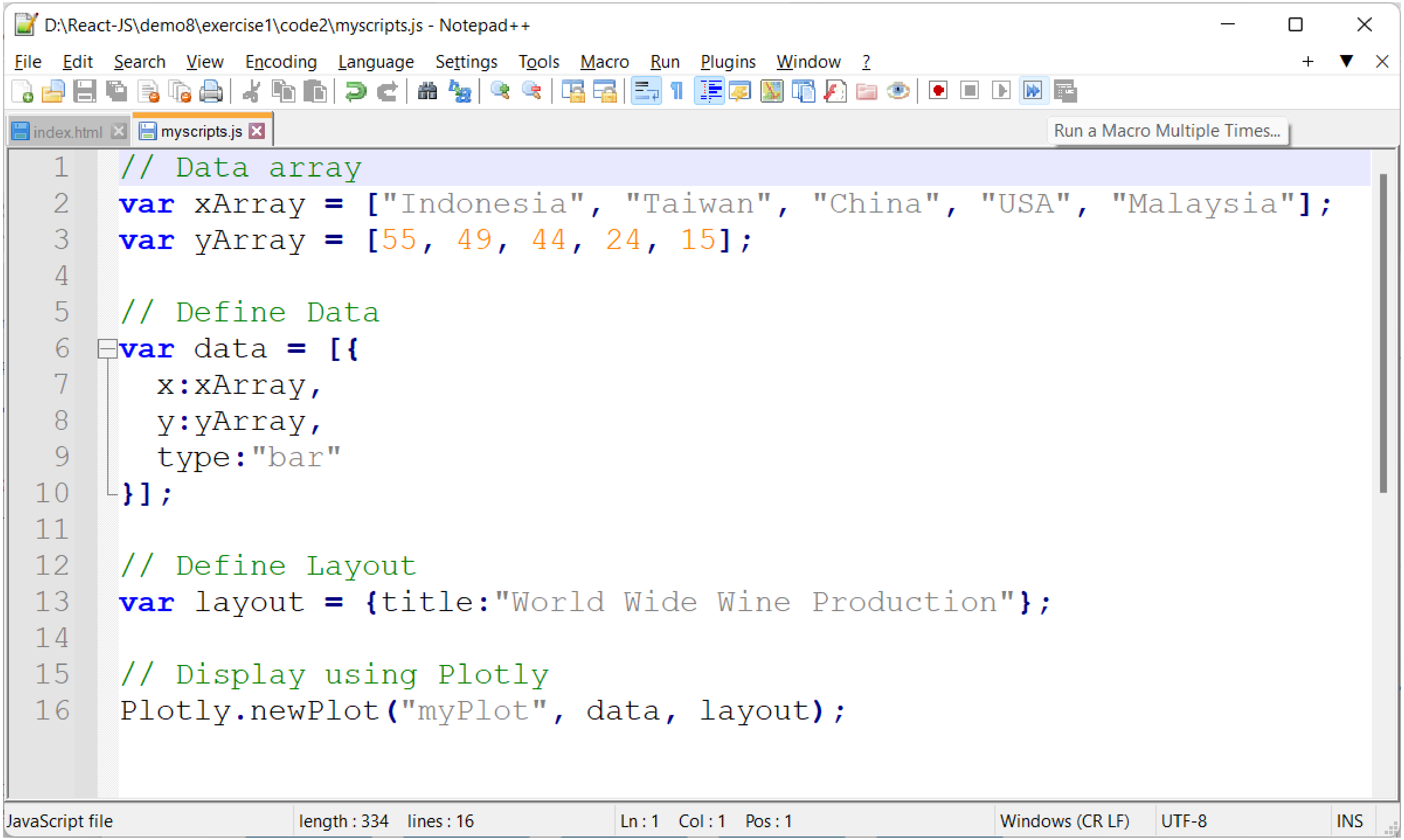
Write Code *in* HTML File


D:\React-JS\demo8\exercise1\code2\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js

```html
<!DOCTYPE html>
<html>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<body>
    <h1>Draw Bar Charts with Plotly.js</h1>
    <hr>
    <div id="myPlot" style="width:100%;max-width:700px"></div>

<script type="text/javascript" src="myscripts.js"></script>
</body>
</html>
```

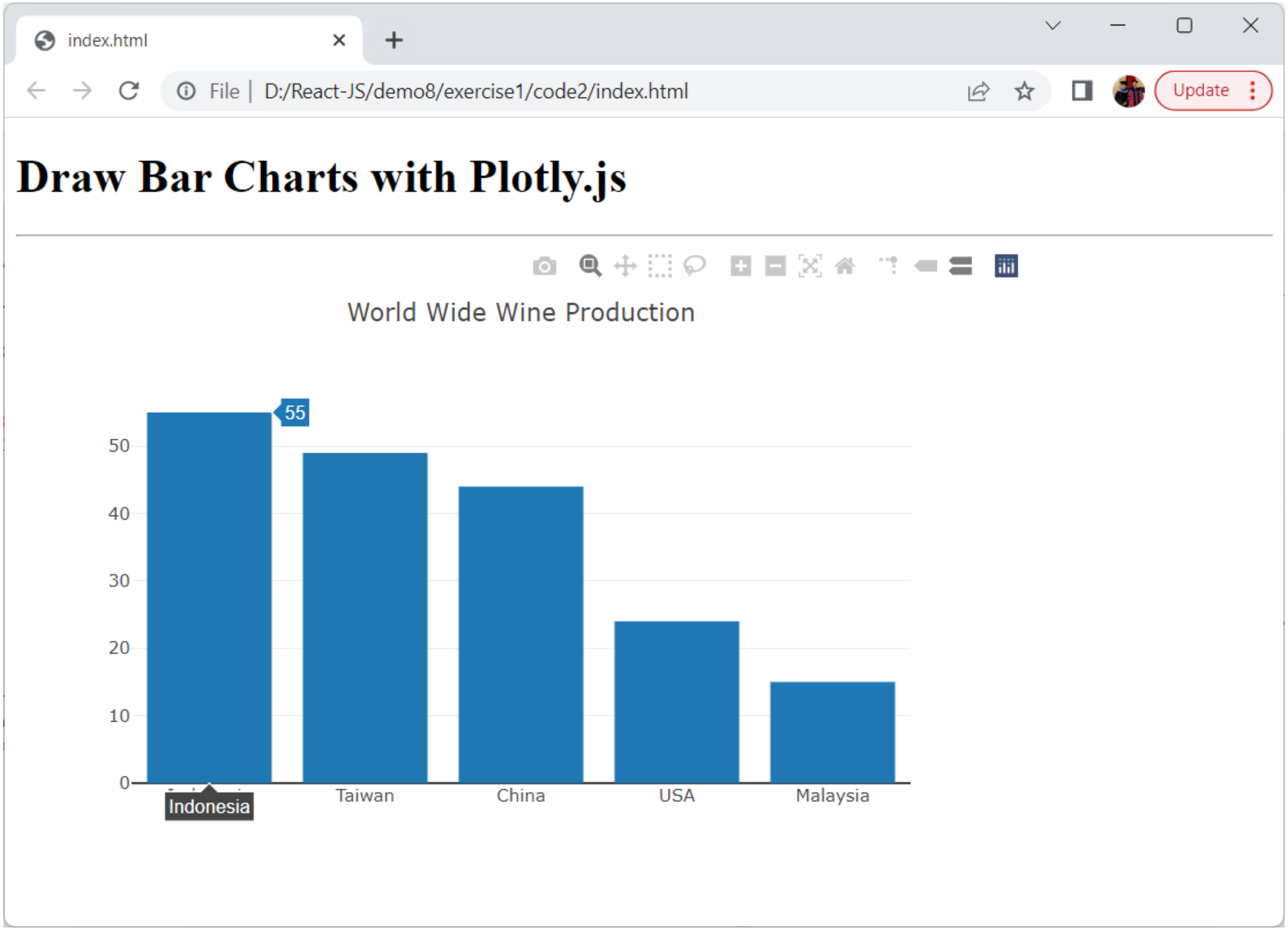
Hyper Text Markup Language file | length : 288 lines : 11 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

Write Code *in* Javascript File


D:\React-JS\demo8\exercise1\code2\myscripts.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js
Run a Macro Multiple Times...
JavaScript file
length : 334 lines : 16
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

```javascript
// Data array
var xArray = ["Indonesia", "Taiwan", "China", "USA", "Malaysia"];
var yArray = [55, 49, 44, 24, 15];

// Define Data
var data = [{
  x:xArray,
  y:yArray,
  type:"bar"
}];

// Define Layout
var layout = {title:"World Wide Wine Production"};

// Display using Plotly
Plotly.newPlot("myPlot", data, layout);
```

Expected Output



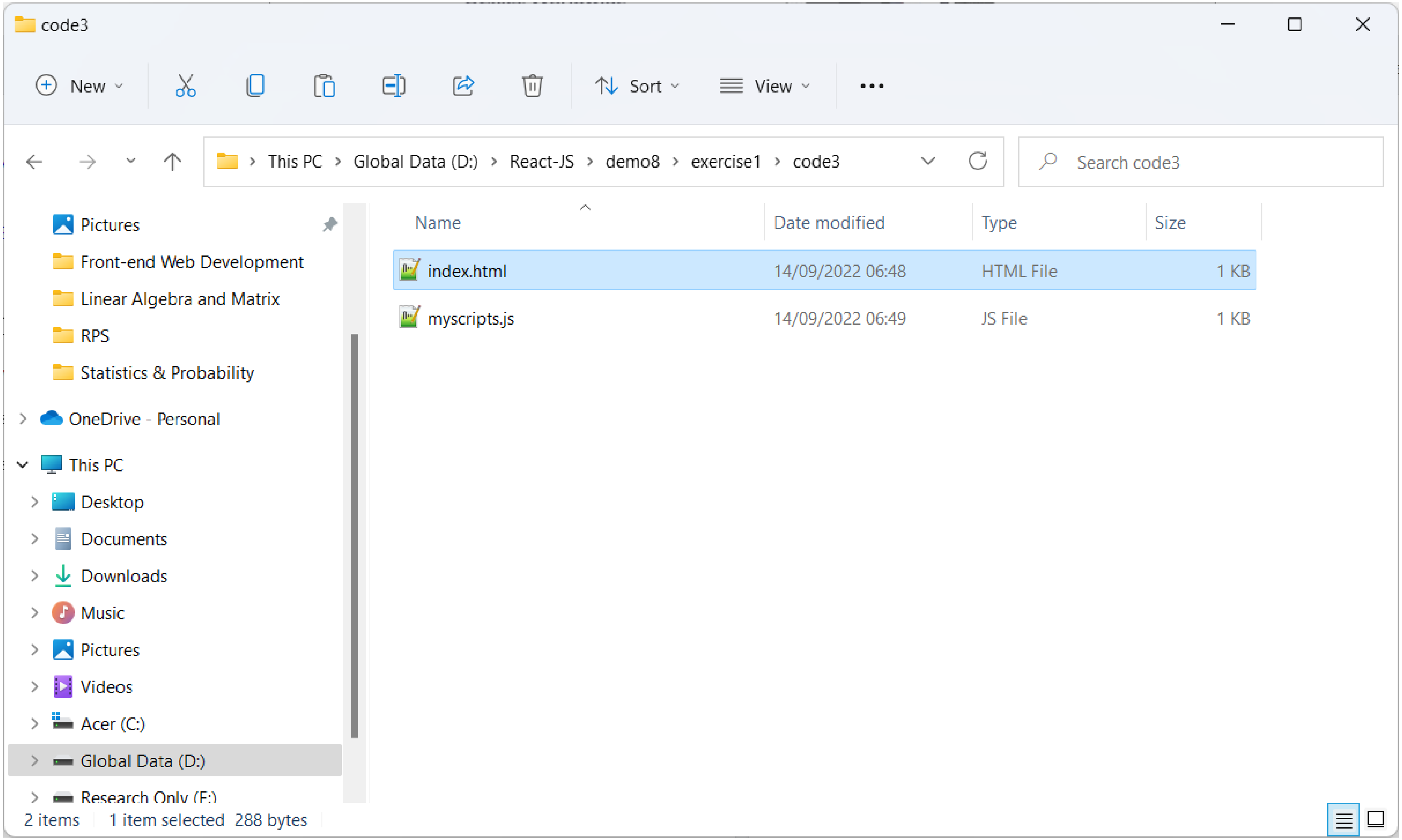
index.html

File | D:/React-JS/demo8/exercise1/code2/index.html

Draw Bar Charts with Plotly.js

Code #3
(Donut Charts)

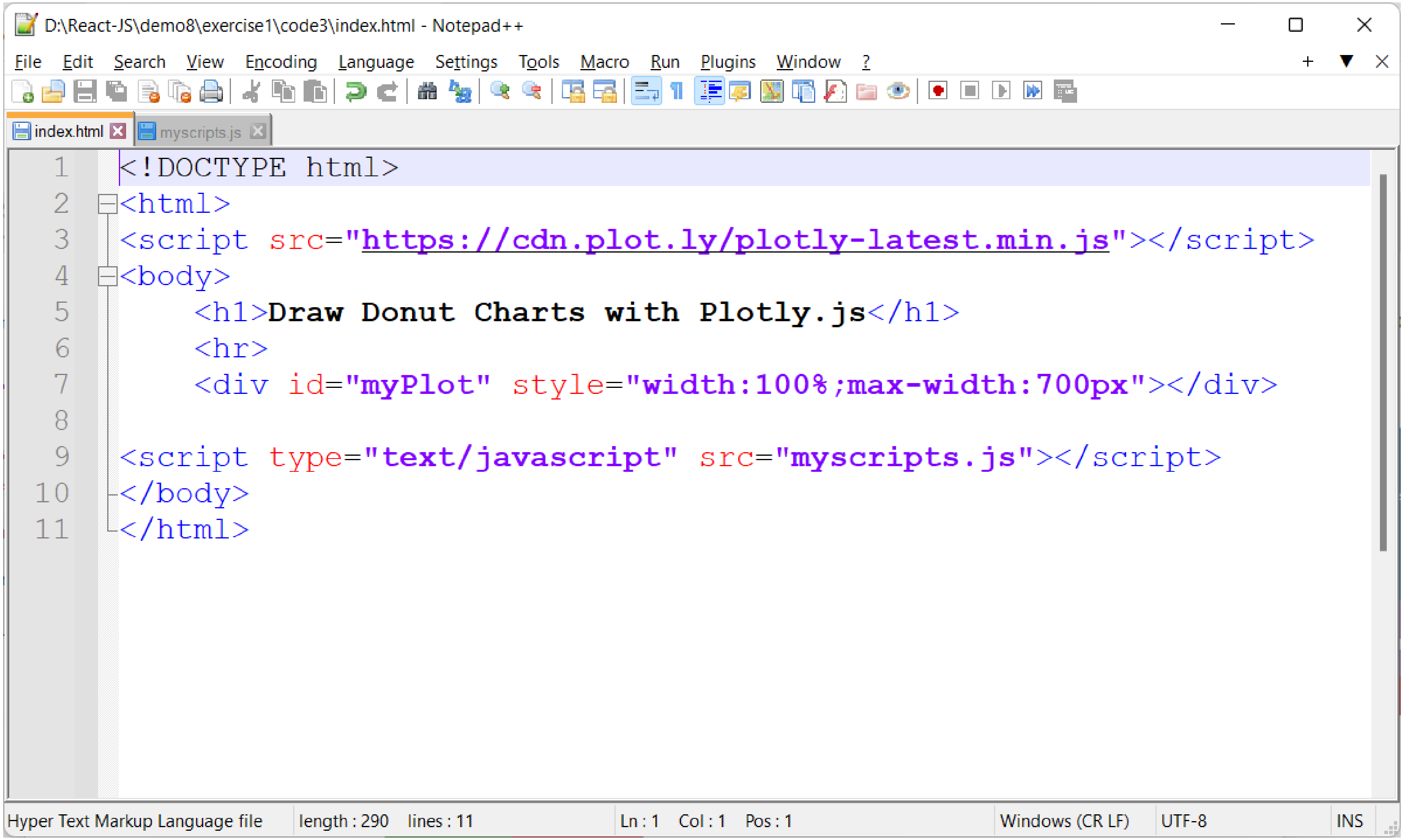
Create HTML & JavaScript Files



Name	Date modified	Type	Size
index.html	14/09/2022 06:48	HTML File	1 KB
myscripts.js	14/09/2022 06:49	JS File	1 KB

2 items 1 item selected 288 bytes

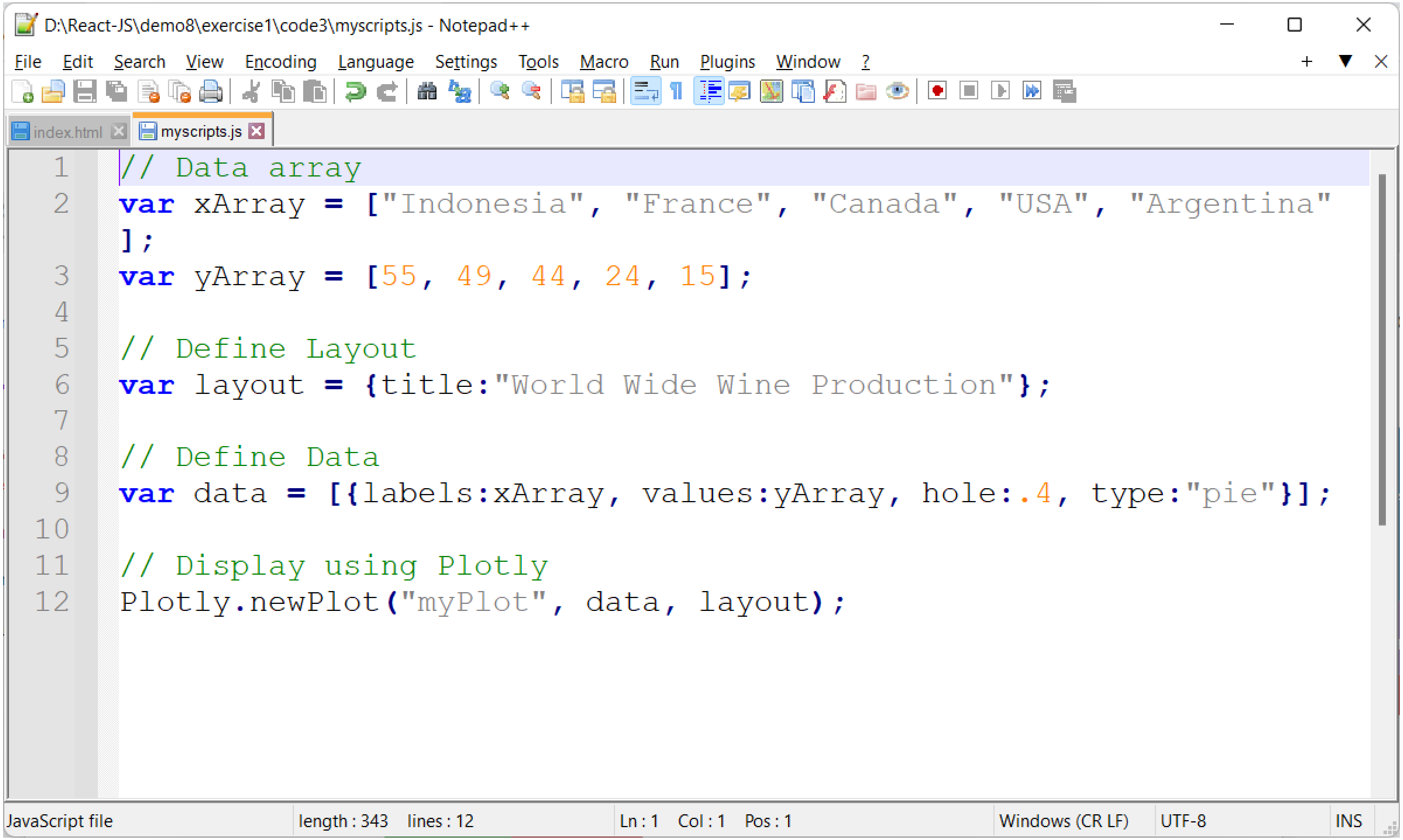
Write Code *in* HTML File


D:\React-JS\demo8\exercise1\code3\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js

```html
<!DOCTYPE html>
<html>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<body>
    <h1>Draw Donut Charts with Plotly.js</h1>
    <hr>
    <div id="myPlot" style="width:100%;max-width:700px"></div>

<script type="text/javascript" src="myscripts.js"></script>
</body>
</html>
```

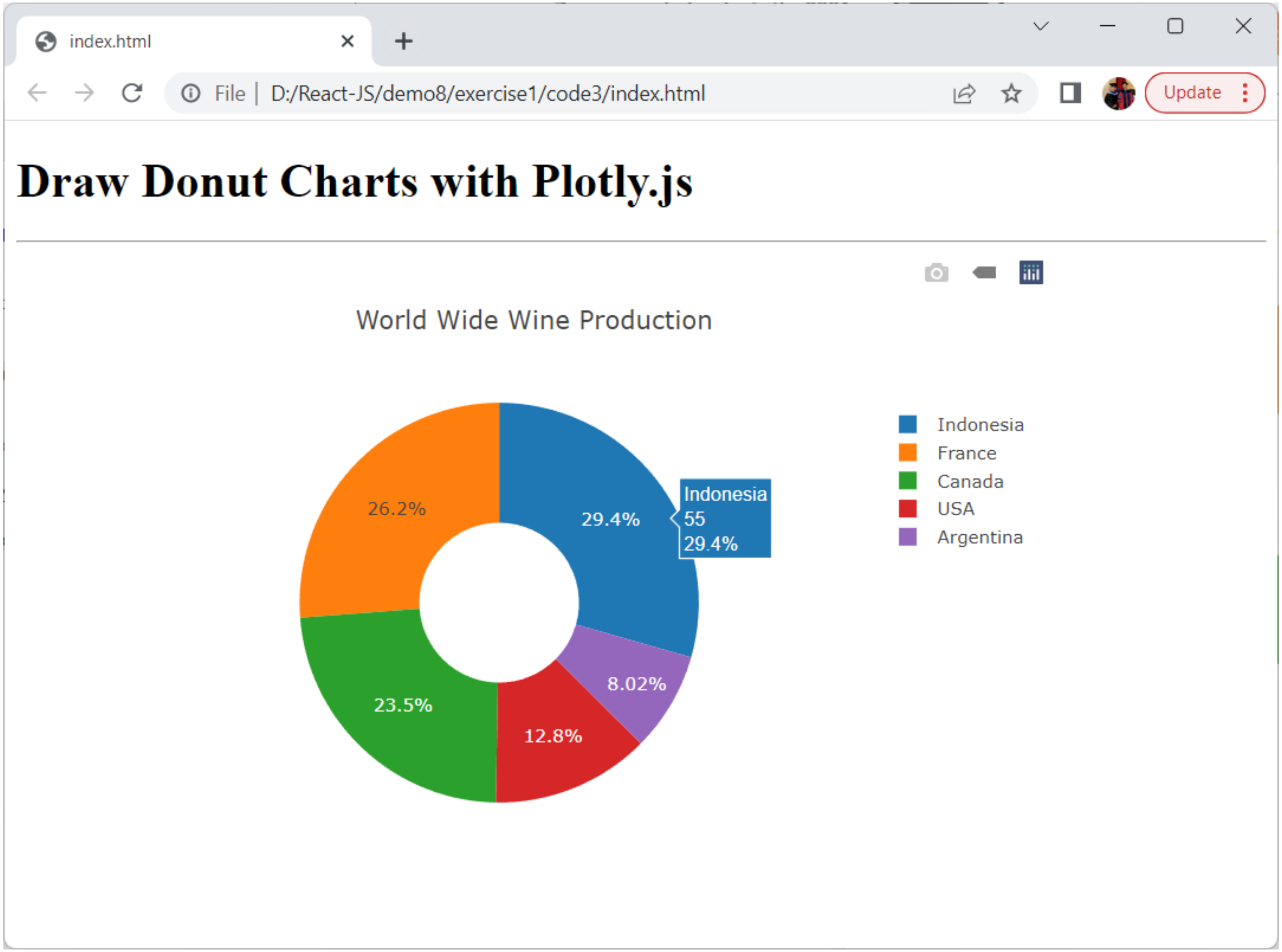
Hyper Text Markup Language file
length : 290 lines : 11
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

Write Code *in* Javascript File



```
// Data array
var xArray = ["Indonesia", "France", "Canada", "USA", "Argentina"];
var yArray = [55, 49, 44, 24, 15];

// Define Layout
var layout = {title:"World Wide Wine Production"};

// Define Data
var data = [{labels:xArray, values:yArray, hole:.4, type:"pie"}];

// Display using Plotly
Plotly.newPlot("myPlot", data, layout);
```

Expected Output

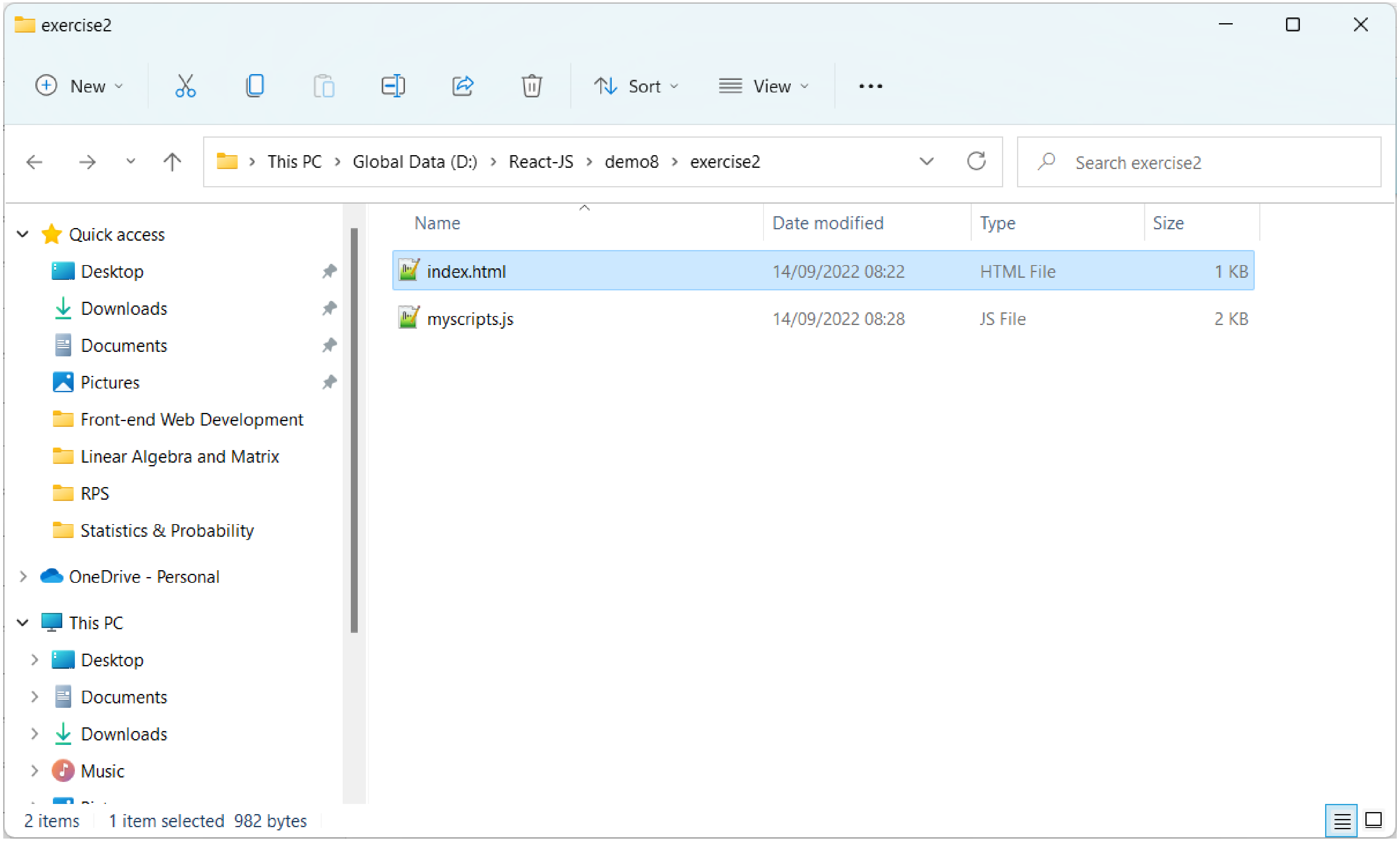


Draw Donut Charts with Plotly.js

Exercise #2

Draw Dynamic Donut Charts with Plotly.js

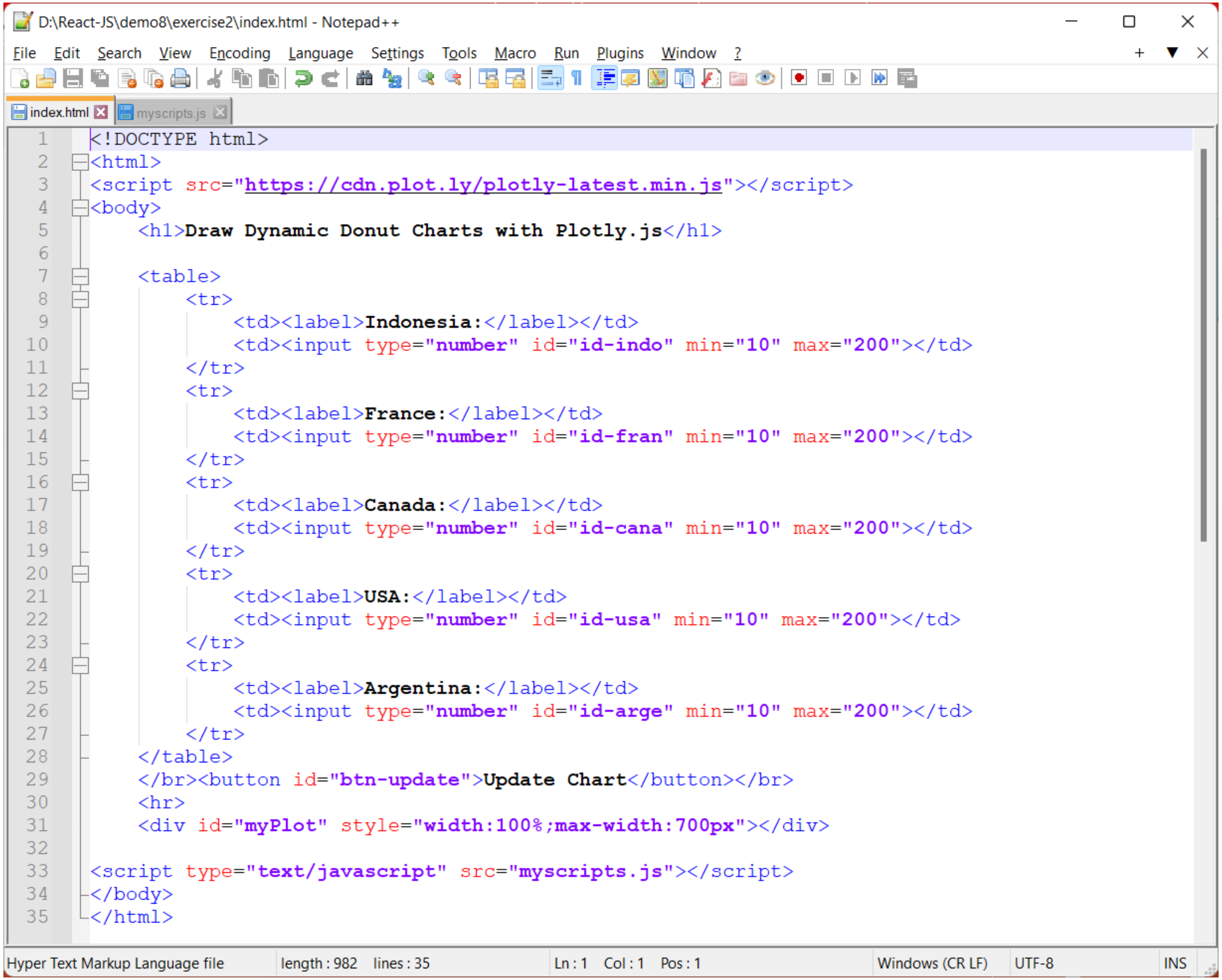
Create HTML & JavaScript Files



Name	Date modified	Type	Size
index.html	14/09/2022 08:22	HTML File	1 KB
myscripts.js	14/09/2022 08:28	JS File	2 KB

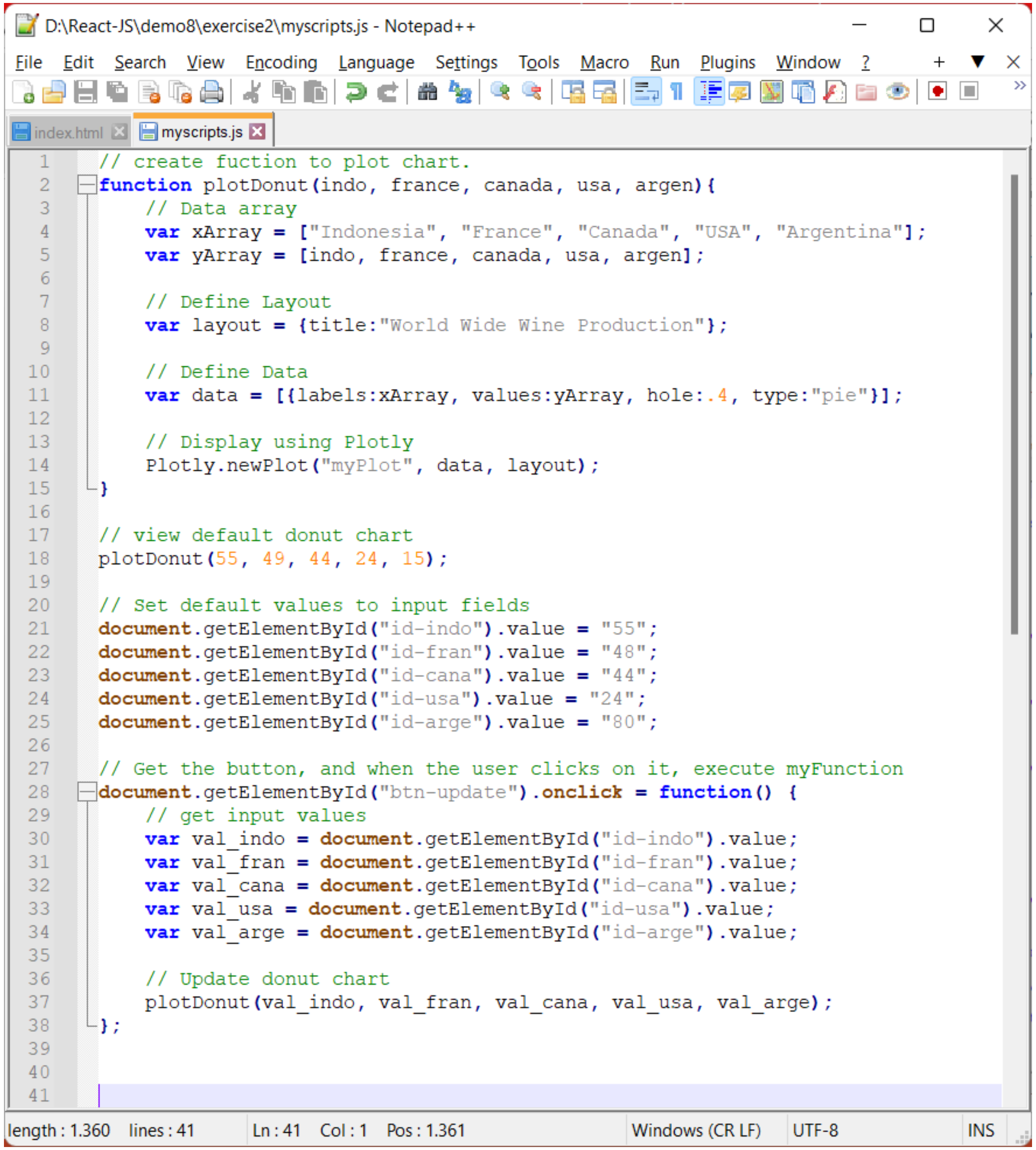
2 items 1 item selected 982 bytes

Write Code *in* HTML File


D:\React-JS\demo8\exercise2\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js
Hyper Text Markup Language file
length : 982 lines : 35
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

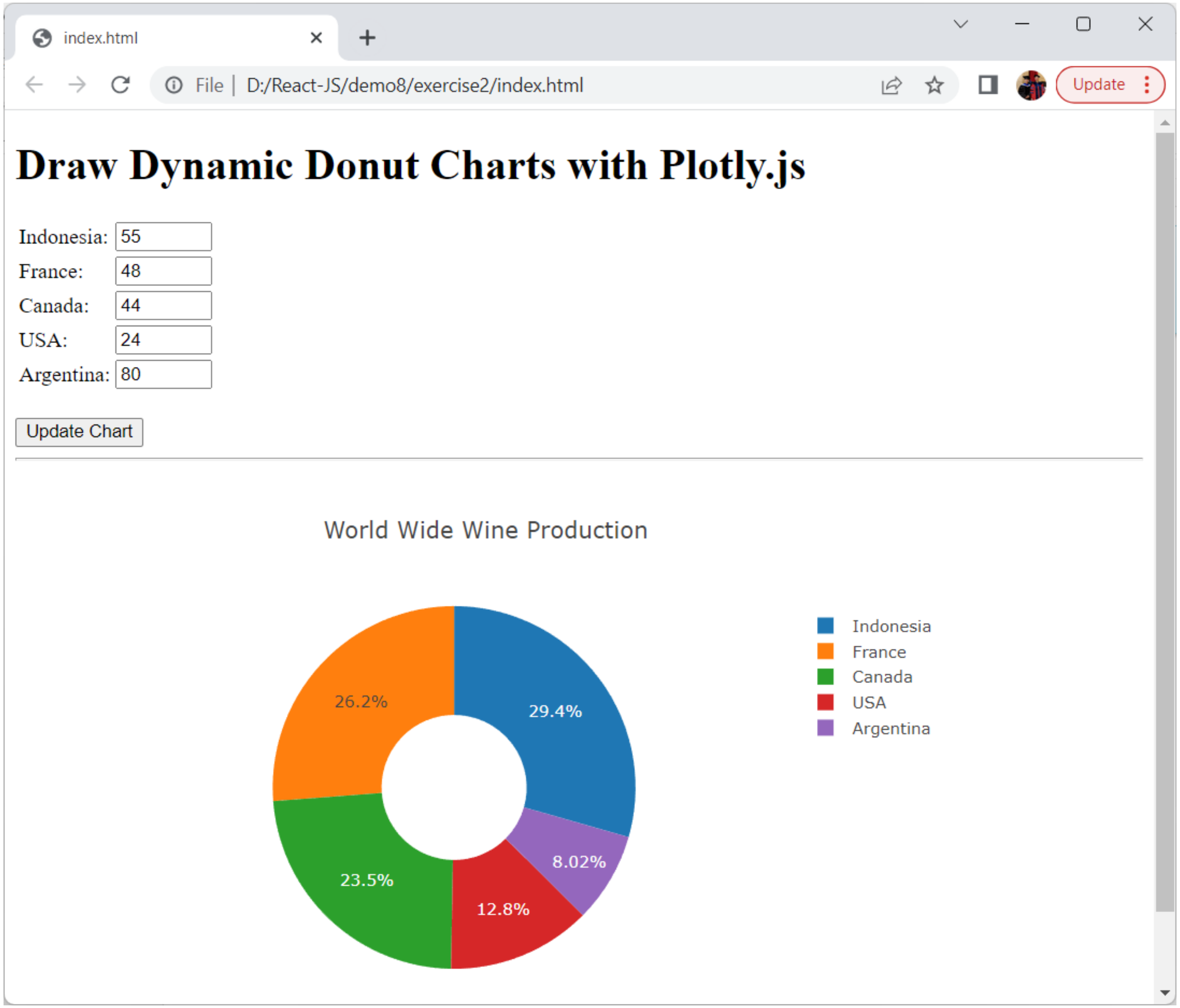
```html
<!DOCTYPE html>
<html>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<body>
    <h1>Draw Dynamic Donut Charts with Plotly.js</h1>

    <table>
        <tr>
            <td><label>Indonesia:</label></td>
            <td><input type="number" id="id-indo" min="10" max="200"></td>
        </tr>
        <tr>
            <td><label>France:</label></td>
            <td><input type="number" id="id-fran" min="10" max="200"></td>
        </tr>
        <tr>
            <td><label>Canada:</label></td>
            <td><input type="number" id="id-cana" min="10" max="200"></td>
        </tr>
        <tr>
            <td><label>USA:</label></td>
            <td><input type="number" id="id-usa" min="10" max="200"></td>
        </tr>
        <tr>
            <td><label>Argentina:</label></td>
            <td><input type="number" id="id-arge" min="10" max="200"></td>
        </tr>
    </table>
    </br><button id="btn-update">Update Chart</button></br>
    <hr>
    <div id="myPlot" style="width:100%;max-width:700px"></div>

<script type="text/javascript" src="myscripts.js"></script>
</body>
</html>
```

Write Code *in* Javascript File


D:\React-JS\demo8\exercise2\myscripts.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html
myscripts.js
length : 1.360 lines : 41
Ln : 41 Col : 1 Pos : 1.361
Windows (CR LF)
UTF-8
INS

```javascript
// create fuction to plot chart.
function plotDonut(indo, france, canada, usa, argen){
    // Data array
    var xArray = ["Indonesia", "France", "Canada", "USA", "Argentina"];
    var yArray = [indo, france, canada, usa, argen];

    // Define Layout
    var layout = {title:"World Wide Wine Production"};

    // Define Data
    var data = [{labels:xArray, values:yArray, hole:.4, type:"pie"}];

    // Display using Plotly
    Plotly.newPlot("myPlot", data, layout);
}

// view default donut chart
plotDonut(55, 49, 44, 24, 15);

// Set default values to input fields
document.getElementById("id-indo").value = "55";
document.getElementById("id-fran").value = "48";
document.getElementById("id-cana").value = "44";
document.getElementById("id-usa").value = "24";
document.getElementById("id-arge").value = "80";

// Get the button, and when the user clicks on it, execute myFunction
document.getElementById("btn-update").onclick = function() {
    // get input values
    var val_indo = document.getElementById("id-indo").value;
    var val_fran = document.getElementById("id-fran").value;
    var val_cana = document.getElementById("id-cana").value;
    var val_usa = document.getElementById("id-usa").value;
    var val_arge = document.getElementById("id-arge").value;

    // Update donut chart
    plotDonut(val_indo, val_fran, val_cana, val_usa, val_arge);
};


```

Expected Output



Draw Dynamic Donut Charts with Plotly.js

Indonesia: 55

France: 48

Canada: 44

USA: 24

Argentina: 80

World Wide Wine Production

END PRESENTATION

Thank you for your attention

Instructor: S – W – T