



D3 IMPLEMENTATION II

Contents

- **Function of Data**
- **Event Handling**
- **Data Binding**
- **Data Loading**

Function of Data

Function of Data

- In the DOM Manipulation, we learned about different DOM manipulation methods in D3 such as `append()`, `style()`, `text()` etc. Each of these functions can take in a constant value or a function **as a parameter**. This function is a function of data. So each of these methods will be called for each of our data values **bound to the DOM**. Consider the following `text()` function.

```
.text(function(d) {  
    return d;  
});
```

- Within this function, we can apply any logic to manipulate the data. These are **anonymous functions**, meaning that there is no name associated with the function. Other than the data (or `d`) parameter, there are two other parameters available to us.

```

<head>
  <meta charset="utf-8">
  <!-- Load d3.js -->
  <!--<script src="./org/d3.min.js"></script>-->
  <script src="https://d3js.org/d3.v6.min.js"></script>

```

```

</head>

```

```

<body>

```

```

  <p></p>

```

```

  <p></p>

```

```

  <p></p>

```

```

<script>

```

```

  var data = [100, 200, 300];

```

```

  var paragraph = d3.select("body")

```

```

    .selectAll("p")

```

```

    .data(data)

```

```

    .text(function (d, i) {

```

```

      console.log("d: " + d);

```

```

      console.log("i: " + i);

```

```

      console.log("this: " + this);

```

```

      return d;

```

```

    });

```

```

</script>

```

```

</body>

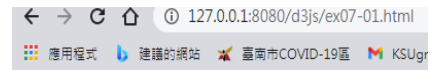
```

```

</html>

```

Try to run
on the server
using XAMPP



100

200

300

Data binding

Elements Console Sources Network Performance Memory Application >>

Page Filesystem >> ex07-01.html x

top
127.0.0.1:8080
d3js
ex07-01.html
d3js.org

```
21 console.log("d: " + d);  
22 console.log("i: " + i);  
23 console.log("this: " + this);  
24  
25 return d;  
26
```

Line 1, Column 1 Coverage: n/a

Pause on caught exceptions
Watch
Breakpoints
No breakpoints
Scope

Console What's New Coverage Issues

top Filter All levels No Issues

9 messages
9 user messages
No errors
No warnings
9 info
ex07-01.html 9
No verbose

☐ Hide network
☒ Preserve log
☒ Selected context only
☒ Group similar messages in console
☐ Log XMLHttpRequests
☐ Eager evaluation
☒ Autocomplete from history
☒ Evaluate triggers user activation

d: 100
i: 0
this: [object HTMLParagraphElement]
d: 200
i: 1
this: [object HTMLParagraphElement]
d: 300
i: 2
this: [object HTMLParagraphElement]

In the above example, the parameter "d" gives you your data element, "i" gives you the index of data in the array and **this** is a reference of the current DOM element. In this case, it is the paragraph element.

<p> </p>

Setting Attribute Value Dynamically

- Along with manipulating DOM elements, we might want to add certain **properties** or **attributes** to our elements. Sometimes, you might want these properties to be bound to or driven by your data.
- Function of data can be useful to setting up properties dynamically based on your data or business logic. For example, if you would like to color your **paragraph** depending on the content of the paragraph, you can do so inside your style property function.

Setting Attribute Value Dynamically

- we have used some logic to check whether the current <p> element's text contains keywords like "Error" or "Warning". If it contains the keyword "Error", we return the color red or if it contains the keyword "Warning", we return the color yellow in **ex07-02.html**.
- **String.prototype.indexOf()**: The **indexOf()** method returns the index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex. Returns -1 if the value is not found.


```
<body>
  <p>Error: This is error.</p>
  <p>Warning:This is warning.</p>
  <script>
    d3.selectAll("p").style("color", function(d, i) {
      var text = this.innerText;

      if (text.indexOf("Error") >= 0) {
        return "red";
      } else if (text.indexOf("Warning") >= 0) {
        return "orange";
      }
    });
  </script>
</body>
```

回報這個執行個體中指定之
字串第一次出現時的所在索引（以零為起始）



Error: This is error.

Warning:This is warning.

Event Handling

Events in D3

- As in all other libraries, D3 also supports built-in events and custom events. We can bind an event listener to any DOM element using `d3.selection.on()` method.
- Syntax is as right:

```
d3.selection.on(type[, listener[, capture]]);
```

- The `on()` method adds an event listener to all selected DOM elements. The **first parameter** is an event type as string such as "click", "mouseover" etc. The **second parameter** is a callback function which will be executed when an event occurs and **the third optional parameter** capture flag may be specified, which corresponds to the W3C `useCapture` flag.



→
After mousing
in the blue
scope



```
<body>
  <div> </div>
  <script>
    d3.selectAll("div")
      .on("mouseover", function() {
        d3.select(this)
          .style("background-color", "orange");

        // Get current event info
        console.log(d3.event);

        // Get x & y co-ordinates
        console.log(d3.mouse(this));
      })
      .on("mouseout", function() {
        d3.select(this)
          .style("background-color", "steelblue");
      });
  </script>
</body>
```

<head>

```
<meta charset="utf-8">
<!-- Load d3.js -->
<!--<script src="./org/d3.min.js"></script>-->
<script src="https://d3js.org/d3.v6.min.js"></script>
<style>
  div {
    height: 100px;
    width: 100px;
    background-color: steelblue;
    margin: 5px;
  }
</style>
```

p.s.

```
// Get x & y co-ordinates
console.log(d3.pointer(this));
```

Data Binding

Data Binding in D3

- You will learn how to bind data to DOM elements and create new elements based on your data. D3 includes the following important methods for data binding.

Method	Description
<code>data()</code>	Joins data to the selected elements
<code>enter()</code>	Creates a selection with placeholder references for missing elements
<code>exit()</code>	Removes nodes and adds them to the exit selection which can be later removed from the DOM
<code>datum()</code>	Injects data to the selected element without computing a join.

data() api

- D3 is data driven. The data() function is used to join the specified array of data to the selected DOM elements and return the updated selection. D3 works with different types of data like *Array*, *CSV*, *TSV*, *JSON*, *XML* etc.
- You can pass two types of value to the data() function, an array of values (number or object) or a function of data.

```
<head>
  <meta charset="utf-8">
  <!-- Load d3.js -->
  <!--<script src="./org/d3.min.js"></script>-->
  <script src="https://d3js.org/d3.v6.min.js"></script>
</head>
```

```
<body>
  <p> </p>
  <p> </p>
  <p> </p>
  <script>
```

```
  var myData = ["Hello World!", "Hello D3", "Hello JavaScript"];
```

```
  var p = d3.select("body")
    .selectAll("p")
    .data(myData)
    .text(function (d, i) {
      return d+"-"+i;
    });
```

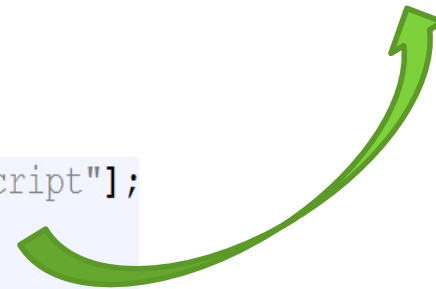
```
</script>
</body>
```

← → ↺ 🏠 ⓘ 127.0.0.1:8080/d3js/ex07-04.html
🌐 應用程式 📖 建議的網站 🚩 臺南市COVID-19區 📧 KSUg

Hello World!-0

Hello D3-1

Hello JavaScript-2



ps `.data(myData)` - the `data()` function then binds our data array 'myData' to the selection returned from the previous selection. Since our selection has single `p` element, the `data()` function will bind the first value from our data array to the `<p>` element.

Data Loading

Data Loading in D3

- In the previous sections, we have worked with data stored in local variables. In this chapter, we will learn to load data from different types of files and bind it to DOM elements.
- D3 can handle different types of data defined either locally in variables or from external files.

Method	Description
<code>d3.csv()</code>	Sends http request to the specified url to load .csv file or data and executes callback function with parsed csv data objects.



Data Loading

- We will learn to load data from csv type and bind it to DOM elements.
- d3.csv(): We can load a csv file or csv data using d3.csv() method.
- **The first parameter** is the url of .csv file, or webapi, or webservice which will return csv data. **The second optional parameter** is a conversion function which allows us to change the representation. **The third optional parameter** is a callback function which will be executed once the .csv file/data is loaded. It passes parsed data object as a parameter to callback function.

```
d3.csv(url[, row, callback]);
```

```

<head>
  <meta charset="utf-8">
  <!-- Load d3.js -->
  <!--<script src="./org/d3.min.js"></script> -->
  <script src="https://d3js.org/d3.v6.min.js"></script>
</body>
<script>
var d=[];
  document.write("This is a test!<br>");
d3.csv("employees.csv", function(error, d) {
  document.write(d.length + "<br>");
if(error){
  console.log(error);
}
else {
for (var i = 0; i < d.length; i++) {
  console.log(d[i].Name);
  console.log(d[i].Age);
}
}
});
</script>
</body>

```



This is a test!

本機磁碟 (C:) > xampp > htdocs > d3js

列印

名稱	修改日期	類型	大小
org	2021/6/6 下午 01:24	檔案資料夾	
others	2021/6/6 下午 01:24	檔案資料夾	
<input checked="" type="checkbox"/> employees.csv	2021/6/6 下午 01:45	Microsoft Excel ...	1 KB
employees.txt	2021/6/5 下午 08:21	文字文件	1 KB
ex01-01-server.html	2021/3/7 下午 05:31	Chrome HTML D...	1 KB
ex04-02t-server.html	2021/2/2 下午 10:53	Chrome HTML D...	1 KB
ex04-02t-server-a.html	2021/6/5 上午 10:24	Chrome HTML D...	1 KB
ex06-03t-server.html	2021/2/2 下午 10:34	Chrome HTML D...	4 KB
ex07-01.html	2021/6/5 上午 11:14	Chrome HTML D...	1 KB
ex07-02.html	2021/6/5 下午 12:54	Chrome HTML D...	1 KB
ex07-03.html	2021/6/5 下午 01:51	Chrome HTML D...	1 KB
ex07-04.html	2021/6/5 下午 04:20	Chrome HTML D...	1 KB
<input checked="" type="checkbox"/> ex07-051.html	2021/6/6 下午 01:58	Chrome HTML D...	1 KB
ex07-052.html	2021/6/6 下午 02:33	Chrome HTML D...	1 KB
xyz.csv	2021/6/4 下午 10:13	Microsoft Excel ...	1 KB

ex07-51.html

SVG Transform

- SVG provides options to transform a single SVG shape element or group of SVG elements. SVG transform supports Translate, Scale, Rotate and Skew.

`translate(x,y)`

Defines a 2D translation

The `dy` attribute indicates a shift along the y-axis on the position of an element or its content.

```
<script src="https://d3js.org/d3.v6.min.js"></script>
<style>
```

```
  svg rect {
    fill: orange;
  }
```

```
  svg text {
    fill:white;
    font: 10px sans-serif;
    text-anchor: end;
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<svg class="chart" width="420" height="120">
  <g transform="translate(0,0)">
    <rect width="50" height="19"></rect>
    <text x="47" y="9.5" dy=".35em">5</text>
  </g>
  <g transform="translate(0,20)">
    <rect width="100" height="19"></rect>
    <text x="97" y="9.5" dy=".35em">10</text>
  </g>
  <g transform="translate(0,40)">
    <rect width="120" height="19"></rect>
    <text x="117" y="9.5" dy=".35em">12</text>
  </g>
```

```
</svg>
```

```
</body>
```

← → ↺ ↻ ⓘ 127.0.0.1/d3js/ex07-06.html

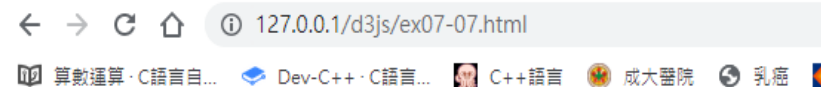
算數運算 · C語言自... Dev-C++ · C語言... C++語言



transform 是變形的效果, 此題須transform.

```
<script src="https://d3js.org/d3.v6.min.js"></script>
<style>
  svg rect {
    fill: orange;
  }

  svg text {
    fill:white;
    font: 10px sans-serif;
    text-anchor: end;
  }
</style>
</head>
<body>
```



```
<script>
```

```
var data = [5, 10, 12, 20, 8];
var width = 200,
scaleFactor = 10,
barHeight = 20;

var graph = d3.select("body")
    .append("svg")
    .attr("width", width) // 寬度
    .attr("height", barHeight * data.length); // 長度

var bar = graph.selectAll("g")
    .data(data)
    .enter()
    .append("g")
    .attr("transform", function(d, i) {
        return "translate(0," + i * barHeight + ")";
    });

bar.append("rect")
    .attr("width", function(d) {
        return d * scaleFactor;
    })
    .attr("height", barHeight - 1);

bar.append("text")
    .attr("x", function(d) { return (d * scaleFactor); })
    .attr("y", barHeight / 2)
    .attr("dy", ".35em")
    .text(function(d) { return d; });
```

```
</script>
```

```
</body>
```

```
</html>
```




無內文的Pie

```
<svg width="300" height="200"> </svg>
```

```
<script>
```

```
var data = [2, 4, 8, 10];
var svg = d3.select("svg"),
    width = svg.attr("width"),
    height = svg.attr("height"),
    radius = Math.min(width, height) / 2,
    g = svg.append("g").attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");
var color = d3.scaleOrdinal(['#4daf4a', '#377eb8', '#ff7f00', '#984ea3', '#e41a1c']);
// Generate the pie
var pie = d3.pie();
// Generate the arcs
var arc = d3.arc()
    .innerRadius(0)
    .outerRadius(radius);
//Generate groups
var arcs = g.selectAll("arc")
    .data(pie(data))
    .enter()
    .append("g")
    .attr("class", "arc")
```

```
//Draw arc paths
arcs.append("path")
    .attr("fill", function(d, i) {
        return color(i);
    })
    .attr("d", arc);
```

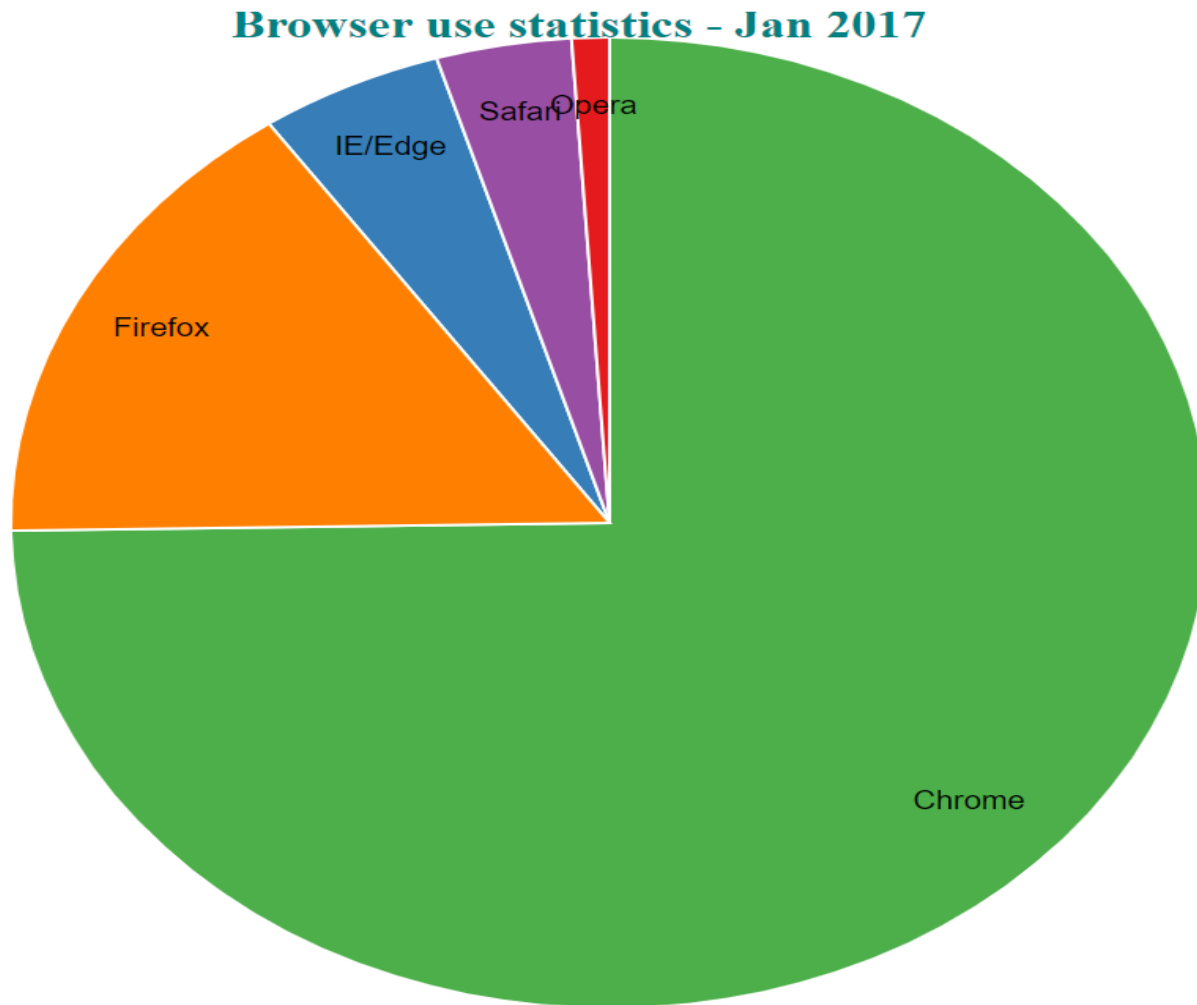
```
</script>
```



- The **<path>** element is the most powerful element in the **SVG** library of basic shapes. It can be used to create lines, curves, arcs, and more. **Paths** create complex shapes by combining multiple straight lines or curved lines.



Read the dataset from CSV for Pie Chart





<head>

```
<!-- Load d3.js -->
```

```
<script src="https://d3js.org/d3.v6.min.js"></script>
```

```
.arc text {
  font: 10px sans-serif;
  text-anchor: middle;
}
```

```
.title {
  fill: teal;
  font-weight: bold;
}
```

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```



```
<body>
  <svg width="500" height="400"></svg>
  <script>
    var svg = d3.select("svg"),
        width = svg.attr("width"),
        height = svg.attr("height"),
        radius = Math.min(width, height) / 2;

    var g = svg.append("g")
        .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

    var color = d3.scaleOrdinal(['#4daf4a', '#377eb8', '#ff7f00', '#984ea3', '#e41a1c']);

    var pie = d3.pie().value(function(d) {
        return d.percent;
    });

    var path = d3.arc()
        .outerRadius(radius - 10)
        .innerRadius(0);

    var label = d3.arc()
        .outerRadius(radius)
        .innerRadius(radius - 80);
```

```
d3.csv("browseruse.csv", function(error, data) {
  if (error) {
    throw error;
  }
  var arc = g.selectAll(".arc")
    .data(pie(data))
    .enter().append("g")
    .attr("class", "arc");

  arc.append("path")
    .attr("d", path)
    .attr("fill", function(d) { return color(d.data.browser); });

  console.log(arc)

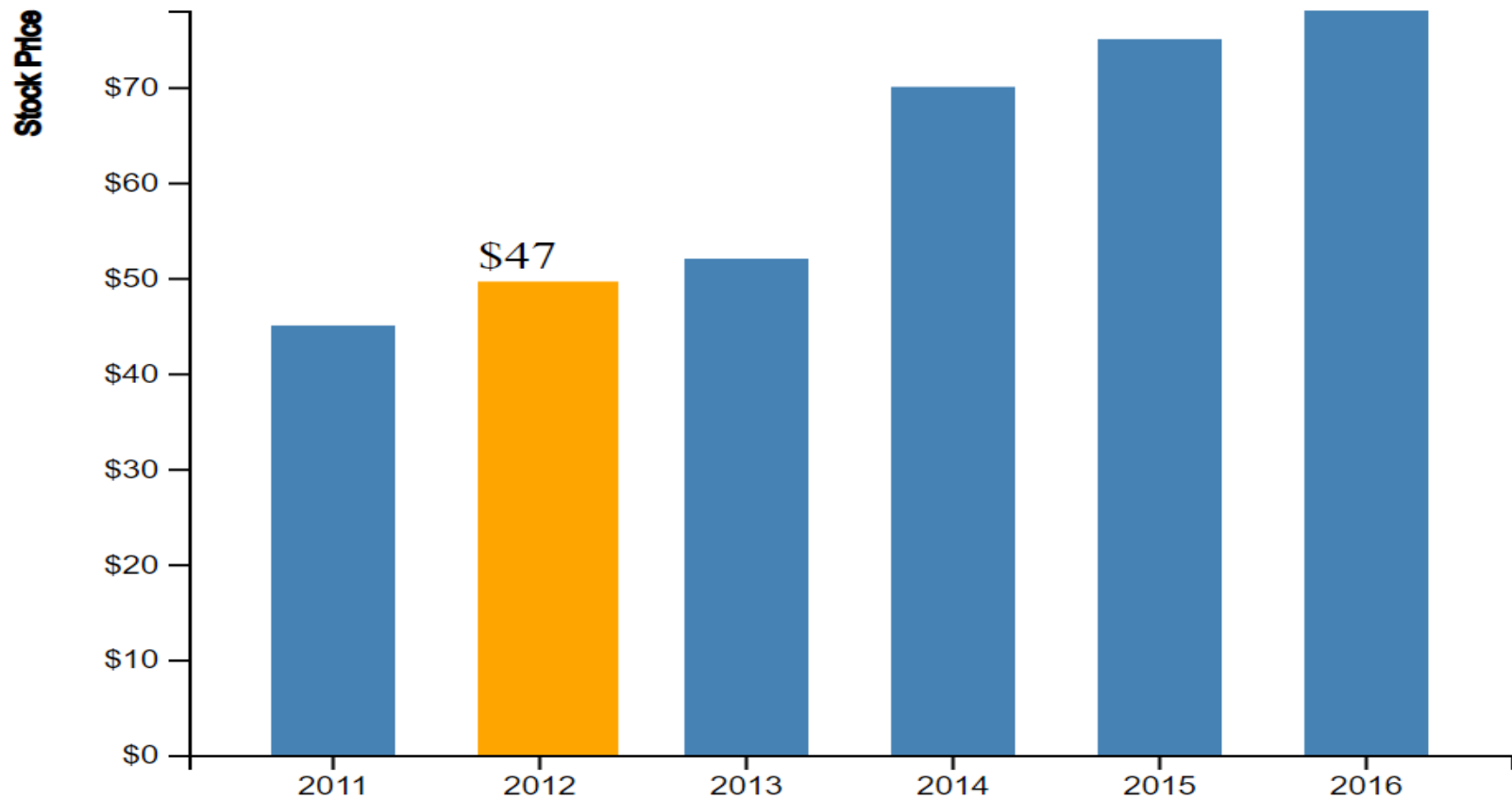
  arc.append("text")
    .attr("transform", function(d) {
      return "translate(" + label.centroid(d) + ")";
    })
    .text(function(d) { return d.data.browser; });
});

svg.append("g")
  .attr("transform", "translate(" + (width / 2 - 120) + "," + 10 + ")")
  .append("text")
  .text("Browser use statistics - Jan 2017")
  .attr("class", "title")
</script>
</body>
</html>
```



Read the dataset from CSV for Bar Chart

XYZ Foods Stock Price





```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .bar {
        fill: steelblue;
      }

      .highlight {
        fill: orange;
      }
    </style>
    <script src="https://d3js.org/d3.v4.min.js"></script>
  </head>
  <body>
    <svg width="600" height="500"></svg>
    <script>
```

```
var svg = d3.select("svg"),
    margin = 200,
    width = svg.attr("width") - margin,
    height = svg.attr("height") - margin;
```



```
svg.append("text")
    .attr("transform", "translate(100,0)")
    .attr("x", 50)
    .attr("y", 50)
    .attr("font-size", "24px")
    .text("XYZ Foods Stock Price")

var x = d3.scaleBand().range([0, width]).padding(0.4),
    y = d3.scaleLinear().range([height, 0]);

var g = svg.append("g")
    .attr("transform", "translate(" + 100 + "," + 100 + ")");
```

```
d3.csv("xyz.csv", function(error, data) {  
  if (error) {  
    throw error;  
  }  
  
  x.domain(data.map(function(d) { return d.year; }));  
  y.domain([0, d3.max(data, function(d) { return d.value; })]);  
  
  g.append("g")  
    .attr("transform", "translate(0," + height + ")")  
    .call(d3.axisBottom(x))  
    .append("text")  
    .attr("y", height - 250)  
    .attr("x", width - 100)  
    .attr("text-anchor", "end")  
    .attr("stroke", "black")  
    .text("Year");  
  
  g.append("g")  
    .call(d3.axisLeft(y).tickFormat(function(d) {  
      return "$" + d;  
    })).ticks(10))  
    .append("text")  
    .attr("transform", "rotate(-90)")  
    .attr("y", 6)  
    .attr("dy", "-5.1em")  
    .attr("text-anchor", "end")  
    .attr("stroke", "black")  
    .text("Stock Price");
```

```
g.selectAll(".bar")
  .data(data)
  .enter().append("rect")
  .attr("class", "bar")
  .on("mouseover", onMouseOver) //Add listener for the mouseover event
  .on("mouseout", onMouseOut)   //Add listener for the mouseout event
  .attr("x", function(d) { return x(d.year); })
  .attr("y", function(d) { return y(d.value); })
  .attr("width", x.bandwidth())
  .transition()
  .ease(d3.easeLinear)
  .duration(400)
  .delay(function (d, i) {
    return i * 50;
  })
  .attr("height", function(d) { return height - y(d.value); });
});
```

//mouseover event handler function

```
function onMouseOver(d, i) {  
  d3.select(this).attr('class', 'highlight');  
  d3.select(this)  
    .transition()      // adds animation  
    .duration(400)  
    .attr('width', x.bandwidth() + 5)  
    .attr("y", function(d) { return y(d.value) - 10; })  
    .attr("height", function(d) { return height - y(d.value) + 10; });  
  
  g.append("text")  
    .attr('class', 'val')  
    .attr('x', function() {  
      return x(d.year);  
    })  
    .attr('y', function() {  
      return y(d.value) - 15;  
    })  
    .text(function() {  
      return [ '$' +d.value]; // Value of the text  
    });  
}
```

```
//mouseout event handler function
function onMouseOut(d, i) {
    // use the text label class to remove label on mouseout
    d3.select(this).attr('class', 'bar');
    d3.select(this)
        .transition()      // adds animation
        .duration(400)
        .attr('width', x.bandwidth())
        .attr("y", function(d) { return y(d.value); })
        .attr("height", function(d) { return height - y(d.value); });

    d3.selectAll('.val')
        .remove()
}
```

</script>

</body>

</html>