arman.heydari@usask.ca

CMPT 384 – Information Visualization

D3.js

Lab 1

Course Instructor: Debajyoti Mondal

Lab Tutorial Instructor : **Arman Heydari**(arman.heydari@usask.ca)

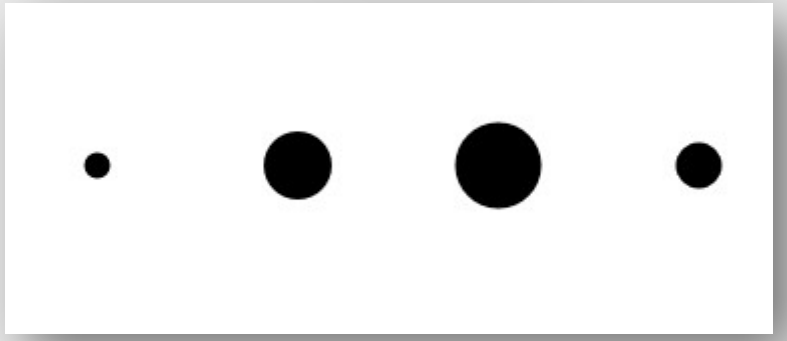
The tutorial attendance is worth 5% of your final grade.

Agenda

- Introduction (short)
- Writing into console in browser
- Random number generation in JS
- Setup server using python
- Read a CSV file (D3)
- D3 basic steps
- D3 chart circle draw

Input.csv

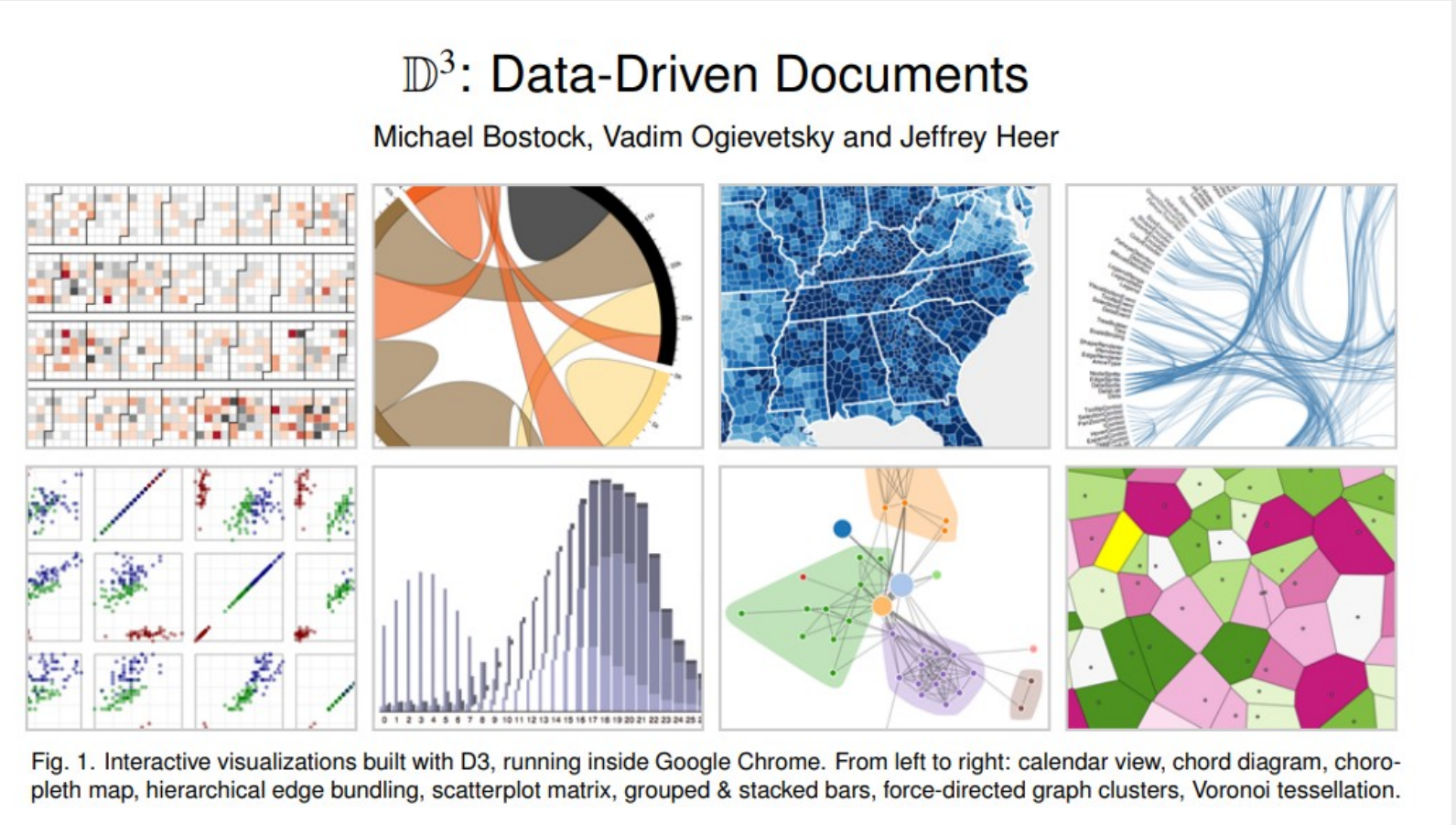
```
team,score
A,13
B,34
C,43
D,23
```



Assumptions …

- Have Heard Of “**World Wide Web**”, **HTML**, **DOM**, **JavaScript** and **CSS**
- Have A **Little Programming Experience** Already
- Aren’t Scared By Unknown Terms such as **CSV**, **SVG**, Or **JSON**
- **WANT To Make Useful, Interactive Visualizations**

D3.JS



IEEE Transactions on Visualization and Computer Graphics
http://vis.stanford.edu/files/2011-D3-InfoVis.pdf

Visualizing with D3

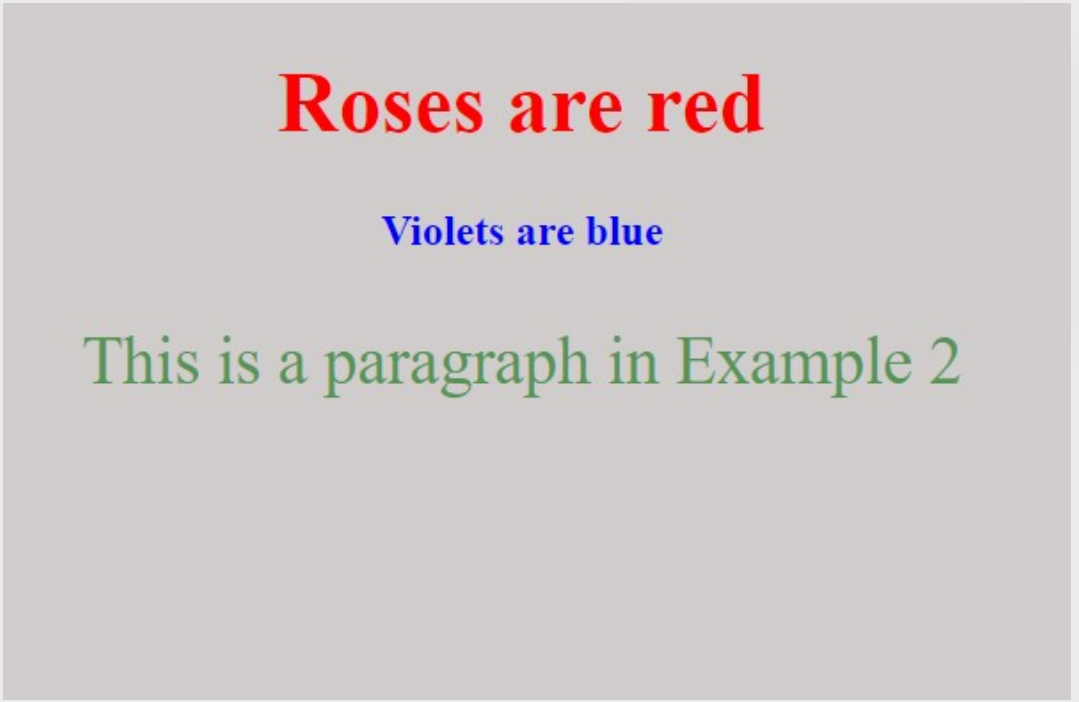
HTML

Describes the content on the page.

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

  <head>
    <meta charset="utf-8">
    <title>Example 1 - Simple HTML</title>
  </head>

  <body>
    <h1> Hello World ! </h1>
    <h6> I am now sentient... </h6>
    <p> This is a new day, a new beginning !</p>
    <p> Things to do -</p>
    <ul>
      <li> Wake up </li>
      <li> Drink Coffee</li>
      <li> Take Over the World</li>
      <li> Watch Netflix</li>
    </ul>
  </body>

</html>
```

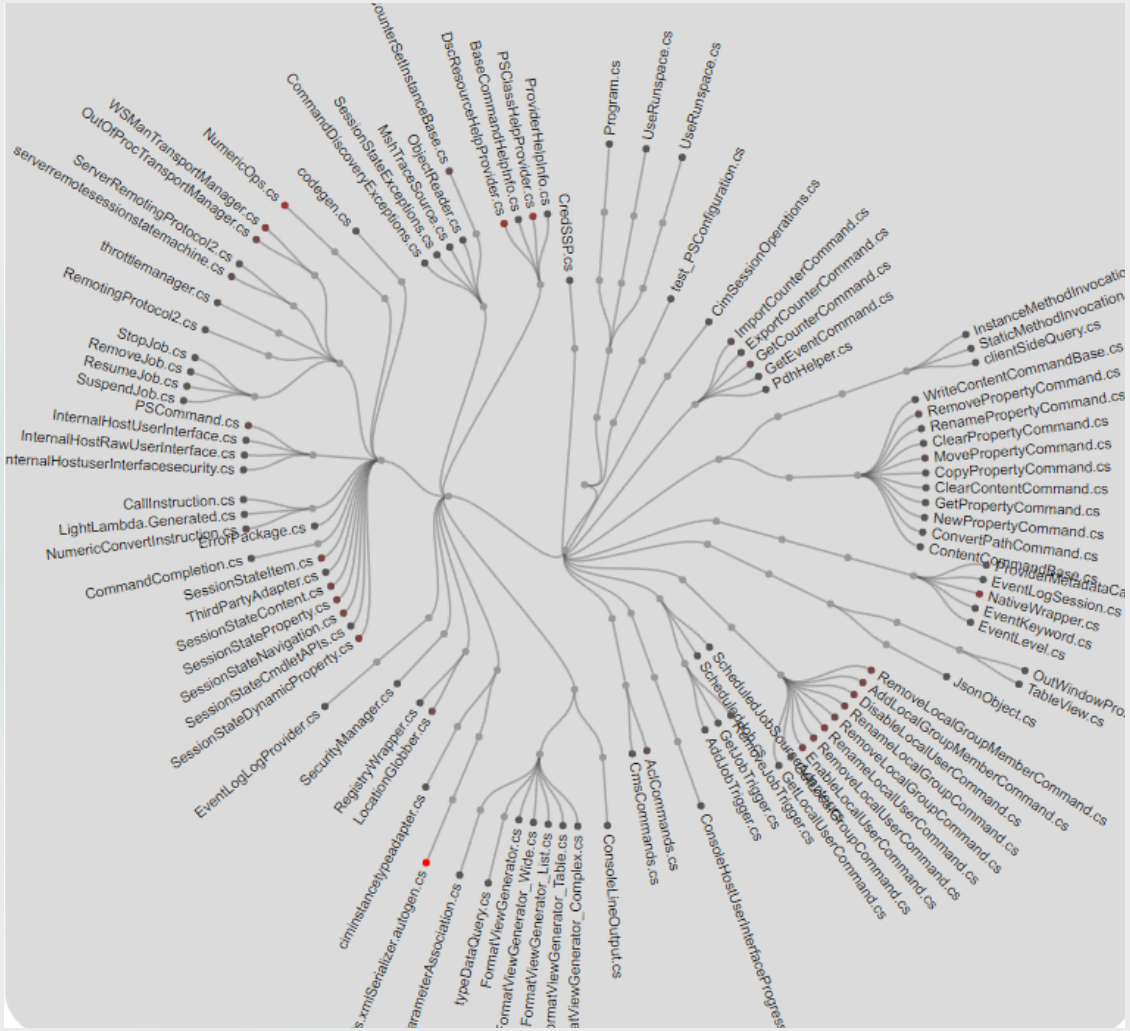
CSS

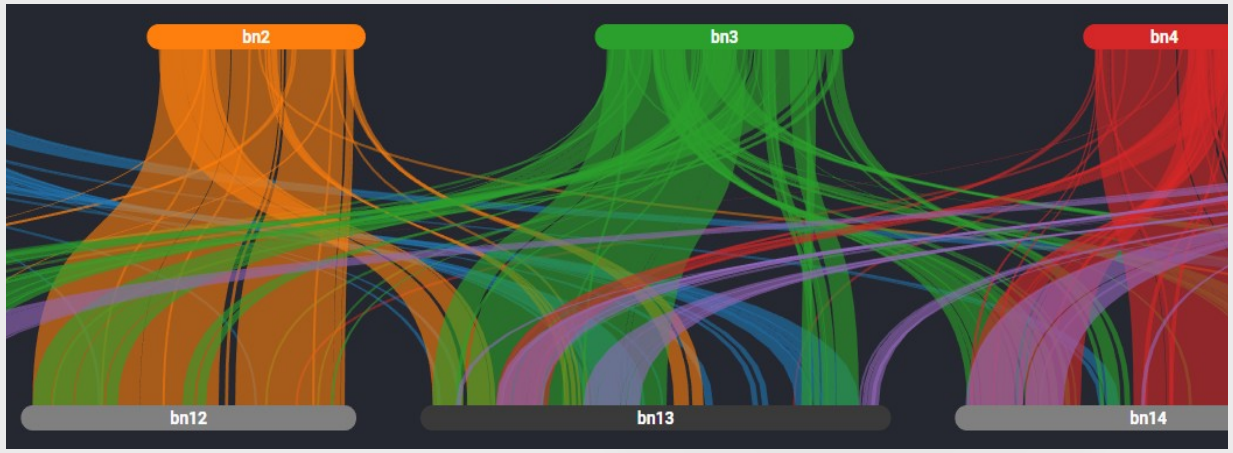
Describes how content should look on a webpage.

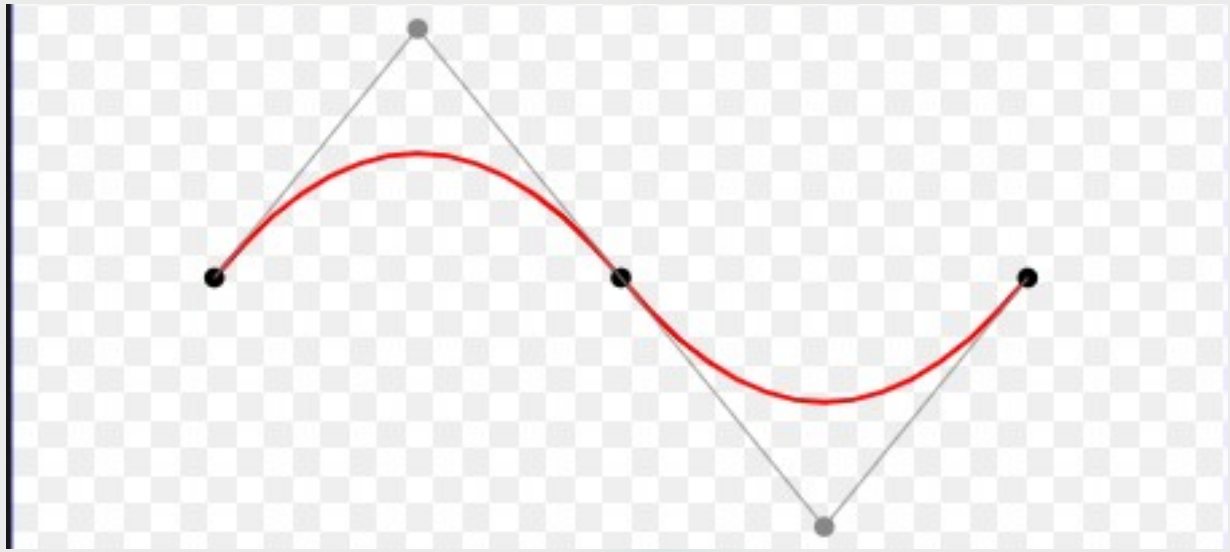


SVG

Vector images defined by points, lines, and curves… mathematical

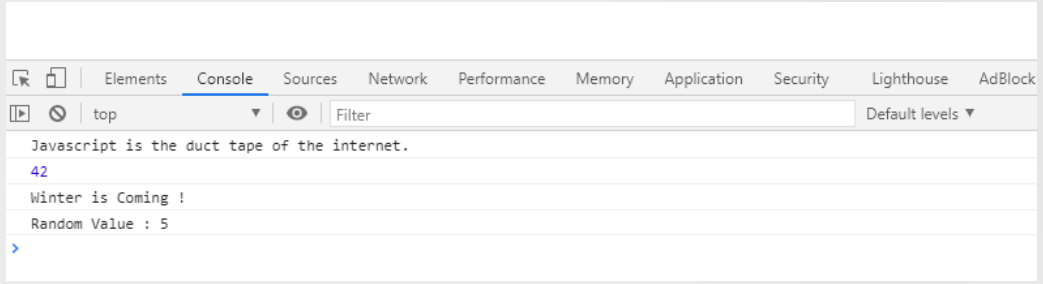






JavaScript – Example 4

```html
<html>

  <head>
    <meta charset="utf-8">
    <title>Example 4 - Javascript</title>
  </head>

  <body>
    <h3> Press F12 to open Developer Tools</h3>
  </body>


  <script type="text/javascript">

      console.log("Javascript is the duct tape of the internet.");

      var data= ["Winter ", "is ", "Coming !"];

      var person = {
          'name':'jon snow',
          'title':'king in the North',
          'description':'Knows nothing'
      };

      var numberStore = 42;

      console.log(numberStore);

      console.log(data[0]+data[1]+data[2]);

      var random_num = Math.floor(Math.random() * 100);
      console.log("Random Value : " + random_num);

  </script>

</html>
```



D3 – Example 5

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <title>Example 5 - Javascript</title>
  </head>

  <body>
    <h3> Press F12 to open Developer Tools</h3>
  </body>

  <script type="text/javascript" src="https://d3js.org/d3.v5.min.js"></script>
  <script type="text/javascript">

    d3.select('body').append("p").text("Bears, Beets, Battlestar Galactica");

  </script>

</html>
```

Method Chaining …

```
// Method Chaining
d3.select('body').append("p").text("Bears, Beets, Battlestar Galactica");
```

```
// Method Chaining
d3.select('body')
.append("p")
.text("Bears, Beets, Battlestar Galactica");
```

```
// Storing each dom element in a variable and then using it
var bodyVariable = d3.select('body');
var newParagraph = bodyVariable.append('p');
newParagraph.text("Bears, Beets, Battlestar Galactica");
```

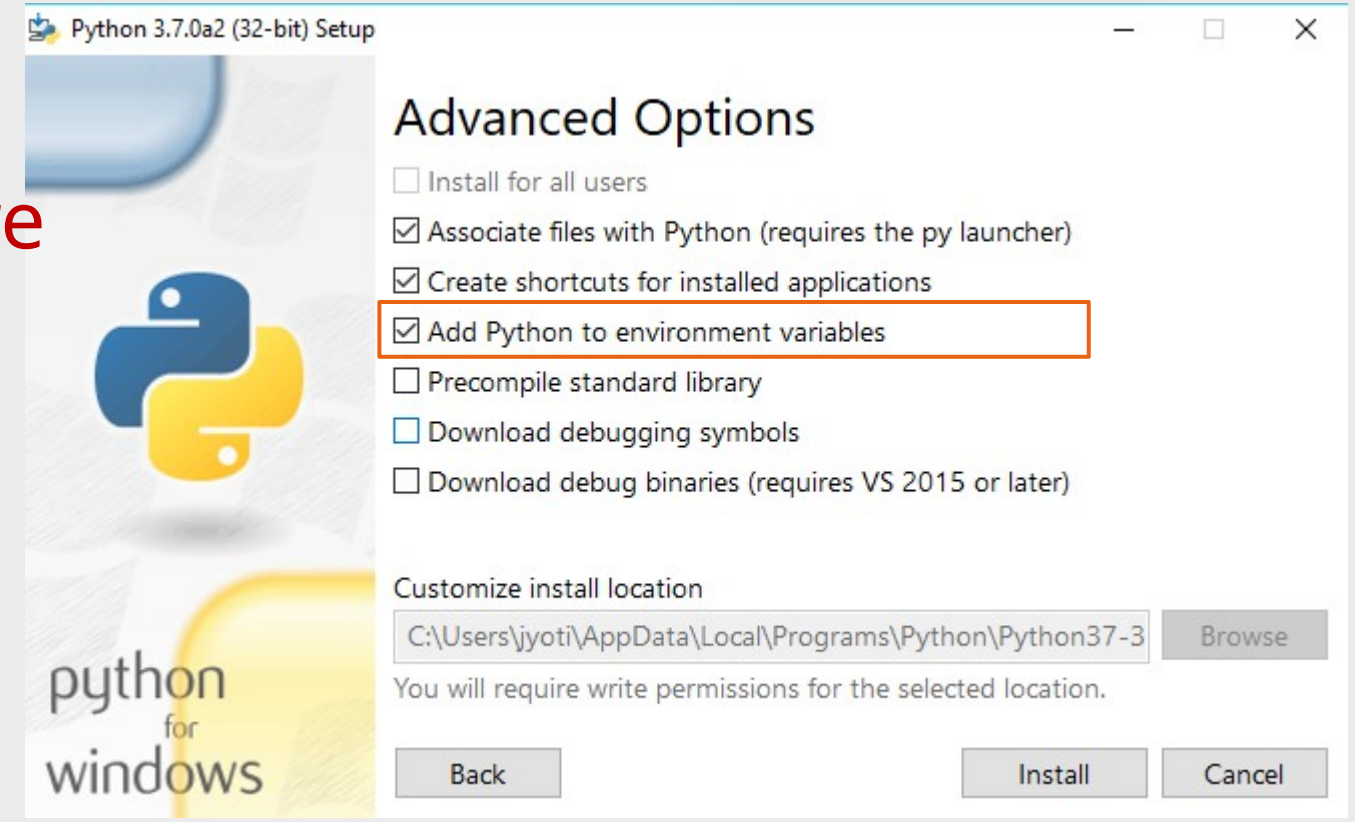
Method chaining lets you write shorter code (and waste less time fretting over variable names)

- Try Modifying the text in the existing h3 tag into “Hello World”

Working with Data …
Frist things First: Create a server

Mac or Linux
- Python is already there
- Open the terminal

Windows
- Install Python
- Open the terminal



Start the Server

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\> python -m SimpleHTTPServer 8888
C:\Users\jyoti\AppData\Local\Programs\Python\Python37-32\python.exe: No
module named SimpleHTTPServer

C:\> python -m http.server 8888
Serving HTTP on 0.0.0.0 port 8888 (http://0.0.0.0:8888/) ...
```

Example 6

```
var dataset;

d3.csv("input.csv").then(function (data) {
    dataset = data;
    generateVisualization();
});
```

- Load Data using csv function in D3
- Store data into global variable *dataset* , that can be accessed anywhere
- Call function *generateVisualization*

Example 6

```
function generateVisualization() {
    d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text(
            function (d) {
                return "Team: " + d.team + ", Score: " + d.score
            }
        );
}
```

```
> dataset
(4) [{…}, {…}, {…}, {…}]
  ▶ 0: {team: "A", score: "13"}
  ▶ 1: {team: "B", score: "34"}
  ▶ 2: {team: "C", score: "43"}
  ▶ 3: {team: "D", score: "23"}
    length: 4
```

Team: A, Score: 13

Team: B, Score: 34

Team: C, Score: 43

Team: D, Score: 23

The D3 enter – update – exit pattern will be explained in more detail in later classes

Creating a SVG

```
// Store SVG width and Height in variables
var width = '500',
    height = '300';
// Create SVG and add to body
var svg = d3.select('body')
    .append('svg')
    .attr('width', width).attr('height', height);
```

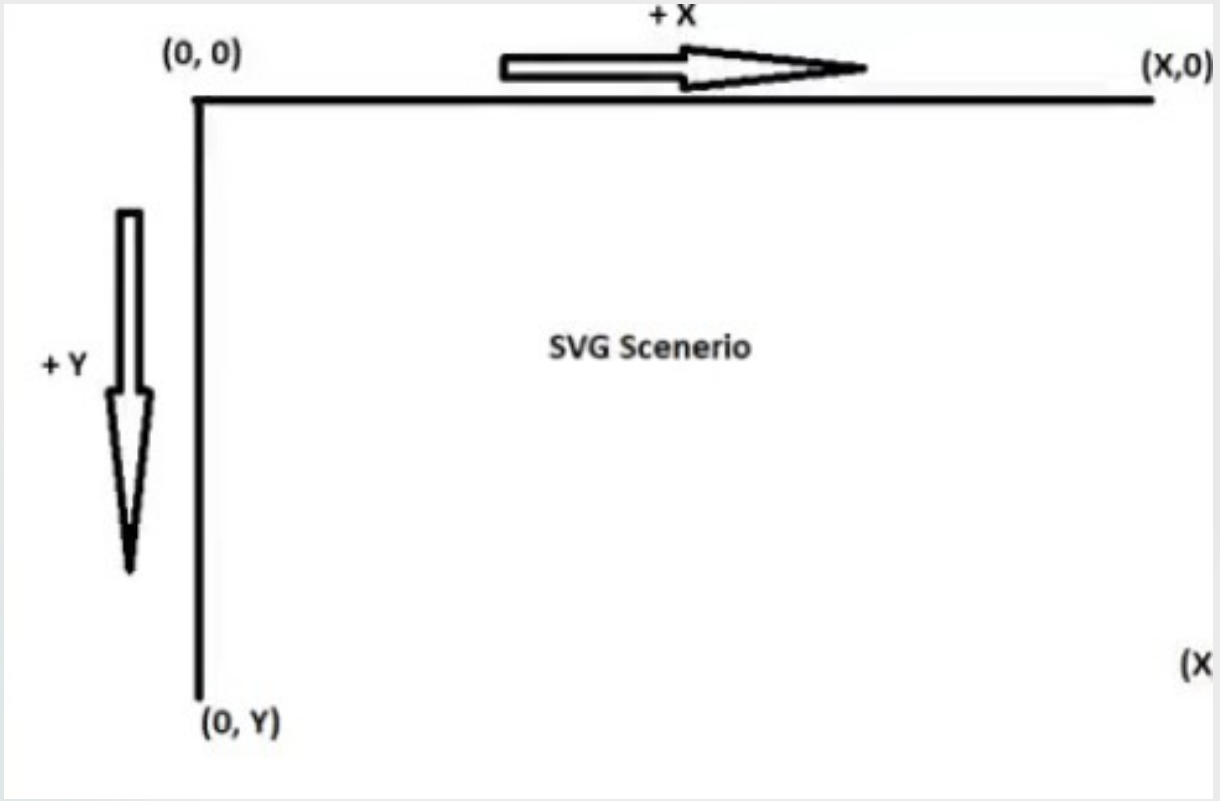
- Store width and height values in variables
- Get the body and attach a new SVG element to it
- Set the width and height attributes
- Store the SVG element in a variable

Creating a Set of Circles

```
// Attach data to virtual array of circles
var circles = svg.selectAll('circle')
    .data(dataset)
    .enter()
    .append('circle')
```

- *selectAll* function gets all the circles on the page
- However, if there aren't any it creates an empty virtual array
- *data* function then binds the elements to the provided data array
- So each value in the array *dataset* is now bound to one virtual element after the enter function
- Then for each virtual element add an actual circle

SVG coordinate system



- The origin (0,0) is located at the top-left corner of the SVG canvas.

Creating a Set of Circles

```
// set circle X position
.attr("cx", function (d, i) {
    return 50 + (i * 100);
})
// set circle Y postion
.attr("cy", 150)
// set radius of Circle
.attr("r", function (d) {
    return d.score;
});
```

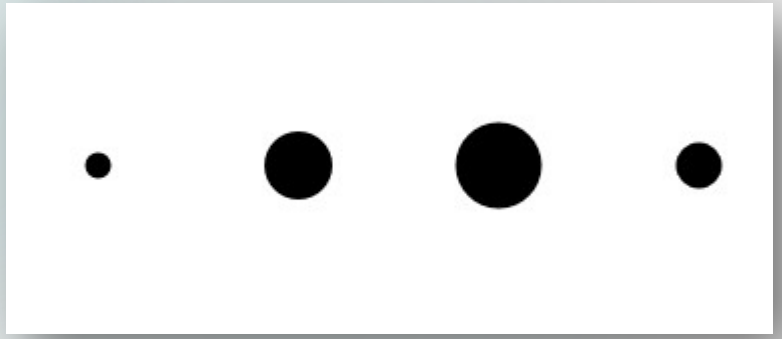
For each circle

- Set the X position based on the array index since circles are laid horizontally
- Set the Y position constant so all circles are in the same level vertically
- Set the radius based on the score, the *d* variable has an object which has both team name and score values, pick only score.

SVG Example

Input.csv

```
team,score
A,13
B,34
C,43
D,23
```



Try changing the code so circles are vertical instead of horizontal.

```
function generateVisualization() {
    // Store SVG width and Height in variables
    var width = '500',
        height = '300';
    // Create SVG and add to body
    var svg = d3.select('body')
        .append('svg')
        .attr('width', width).attr('height', height);
    // Attach data to virtual array of circles
    var circles = svg.selectAll('circle')
        .data(dataset)
        .enter()
        .append('circle')
        // set circle X position
        .attr("cx", function (d, i) {
            return 50 + (i * 100);
        })
        // set circle Y postion
        .attr("cy", 150)
        // set radius of Circle
        .attr("r", function (d) {
            return d.score;
        });
}
```