D3- Data Driven Document

Dr. Bo (Beth) Sun

HTML

```
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>Page Title</h1>
        <p>This is a really interesting paragraph.</p>
    </body>
</html>
```

DOM

- The Document Object Model refers to the hierarchical structure of HTML. Each bracketed tag is an *element.*

<body> Paragraph </body>

- In the HTML above, body is the parent element to both of its children, h1 and p (which are siblings to each other). All elements on the page are descendants of html.

CSS

- Cascading Style Sheets are used to style the visual presentation of HTML pages.

```
body {
    background-color: white;
    color: black;
}
```

CSS

- Selectors

```
h1          /* Selects level 1 headings              */
p           /* Selects paragraphs                    */
.caption    /* Selects elements with class "caption" */
#subnav     /* Selects element with ID "subnav"      */
```

- Rules

```
color: pink;
background-color: yellow;
margin: 10px;
padding: 25px;
```

How to use CSS

```
<head>
    <style type="text/css">
        p {
            font-family: sans-serif;
            color: lime;
        }
    </style>
</head>
```

Directly use it in HTML

```
<head>
    <link rel="stylesheet" href="style.css">
</head>
```

store it in .css file and reference it

JavaScript

```
<body>
    <script type="text/javascript">
        alert("Hello, world!");
    </script>
</body>
```

Directly included in HTML

Stored in separate file and referenced in HTML

```
<head>
    <title>Page Title</title>
    <script type="text/javascript" src="myscript.js"></script>
</head>
```

D3 Basics

```
d3.select("body").append("p").text("New paragraph!");
```

1. Invoked D3's select method, which selects a single element from the DOM using CSS selector syntax. (We selected the body.)
2. Created a new p element and appended that to the end of our selection, meaning just *before* the closing </body> tag in this case.
3. Set the text content of that new, empty paragraph to “New paragraph!”

D3.Chaining.Method

```
d3.select("body").append("p").text("New paragraph!");
```

```
d3.select("body")
    .append("p")
    .text("New paragraph!");
```

what does each function expect and return, the API reference is your friend:
https://github.com/d3/d3/blob/master/API.md

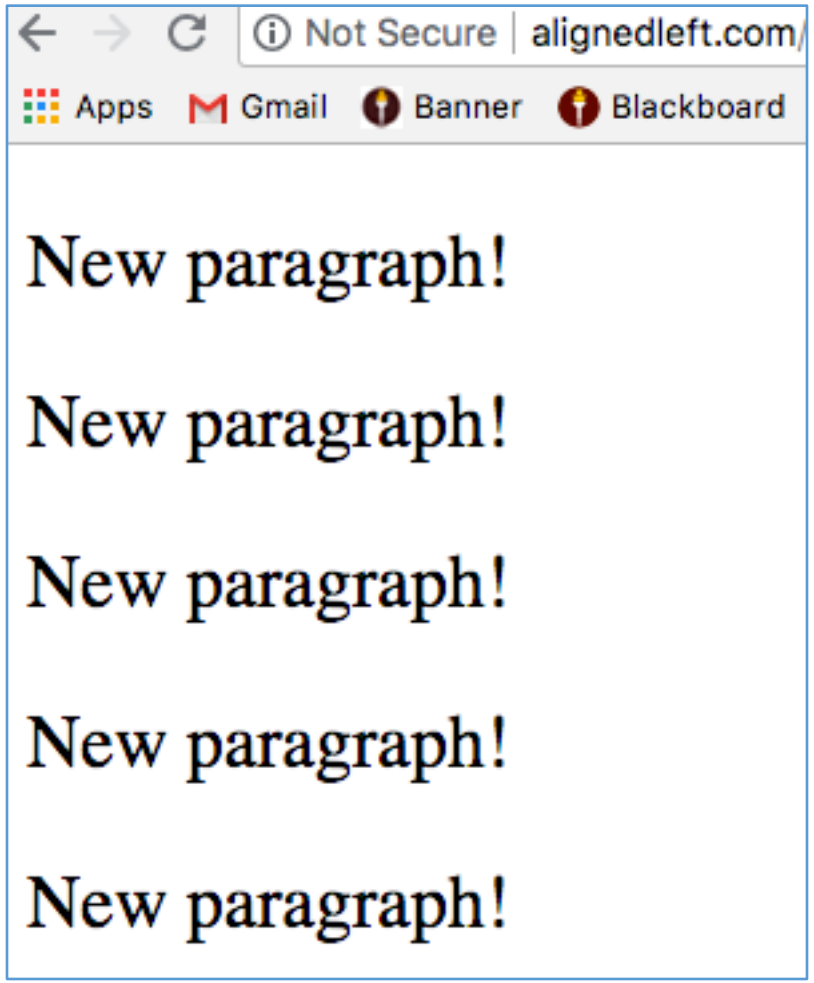
```
d3.select("body")
    .append("p")
    .text("New paragraph!");
```

Chainless

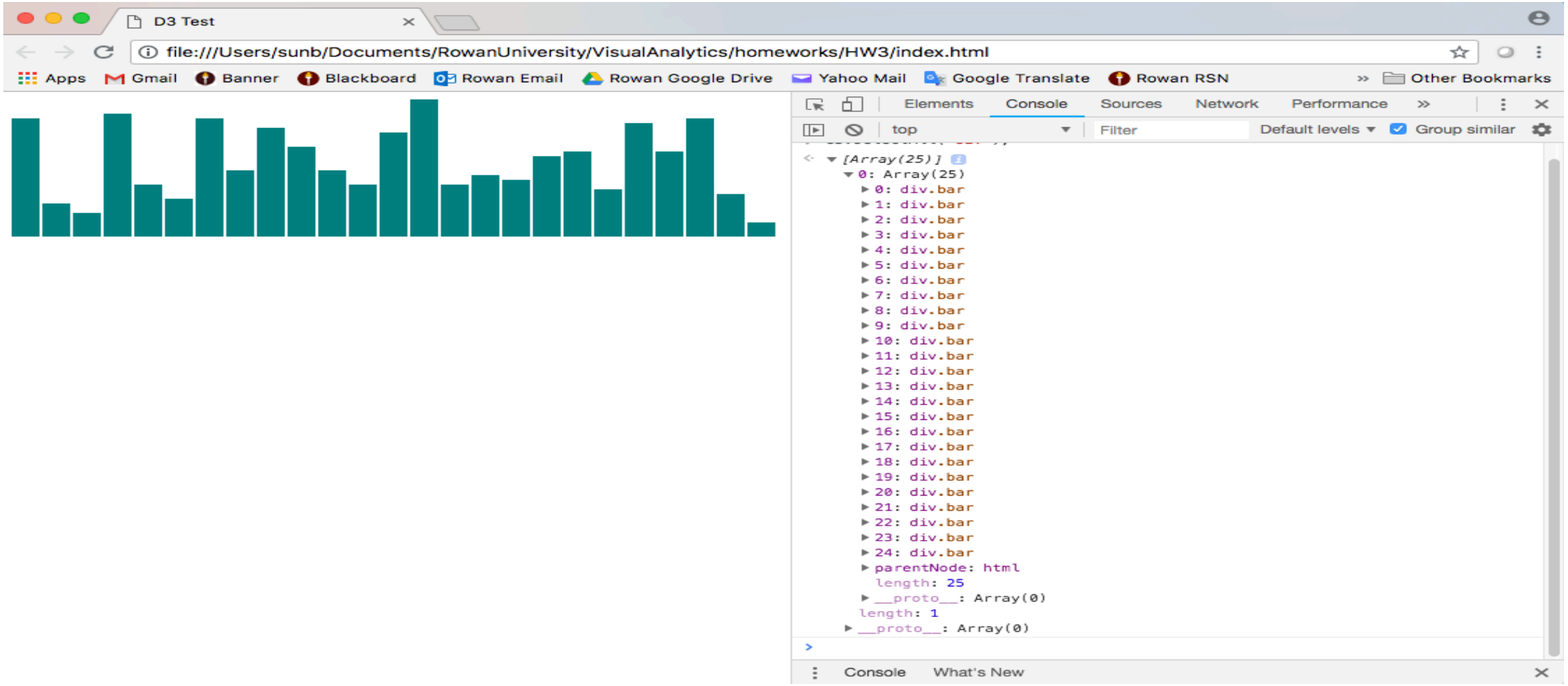
```
var body = d3.select("body");
var p = body.append("p");
p.text("New paragraph!");
```

Data Binding-- selection.data()

```
var dataset = [ 5, 10, 15, 20, 25 ];
d3.select("body").selectAll("p")
    .data(dataset)
    .enter()
    .append("p")
    .text("New paragraph!");
```

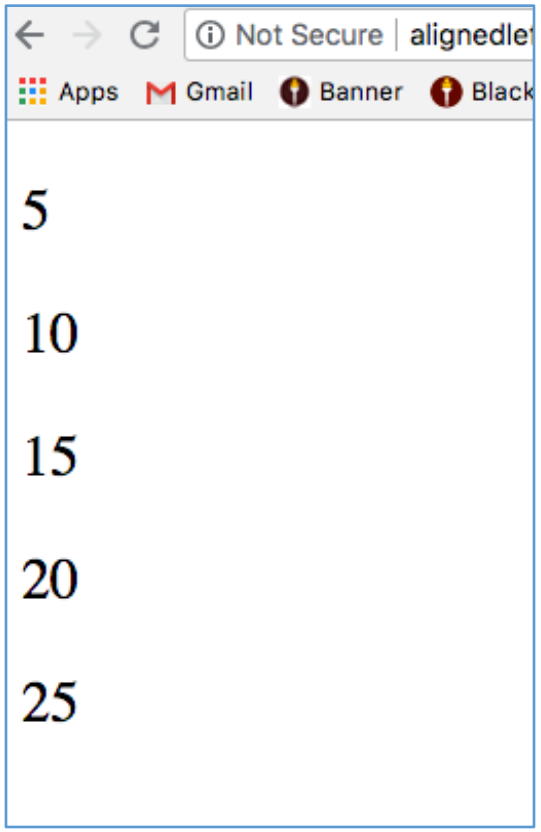


Bound and Determined



Hold Data--function

```
function(input_value) {
    //Calculate something here
    return output_value;
}
```




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

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

Beyond Text--.style()

Not Secure | alignedleft.com/content/03-tutorials/01-d3/70-using-your-data/4.html

Apps Gmail Banner Blackboard Rowan Email Rowan Google Drive Yahoo Mail » Other Bookmarks

I can count up to 5

I can count up to 10

I can count up to 15

I can count up to 20

I can count up to 25

Elements Console Sources Network Performance »

4.html 4.html ×

```
7     </head>
8     <body>
9         <script type="text/javascript">
10
11            var dataset = [ 5, 10, 15, 20, 25 ];
12
13            d3.select("body").selectAll("p")
14                .data(dataset)
15                .enter()
16                .append("p")
17                .text(function(d) {
18                    return "I can count up to " + d;
19                })
20                .style("color", function(d) {
21                    if (d > 15) {   //Threshold of 15
22                        return "red";
23                    } else {
24                        return "black";
25                    }
26                });
27
28        </script>
29    </body>
30 </html>
```

Line 1, Column 1

Drawing Divs

```
<div style="display: inline-block;
            width: 20px;
            height: 75px;
            background-color: teal;"></div>
```

Converted to CSS

```
div.bar {
    display: inline-block;
    width: 20px;
    height: 75px;    /* We'll override this later */
    background-color: teal;
}
```

+

```
<div class="bar">
</div>
```

D3-Setting Attributes -- selection.attr()

```
<p class="caption">
<select id="country">
<img src="logo.png" width="100px" alt="Logo" />
```

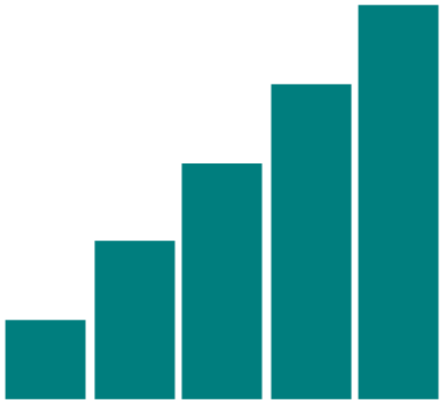
class	caption
id	country
src	logo.png
width	100px
alt	Logo

D3:
.attr ("class", "bar")

Bar Graph!

Not Secure | alignedleft.com/content/03-tutorials/01-d3/80-drawing-divs/3.html

Apps Gmail Banner Blackboard Rowan Email Rowan Google Drive Yahoo Mail Google Translate Rowan RSN Rowan Calenda



Elements Console Sources Network Performance Memory Application Security Audits

3.html 3.html ×

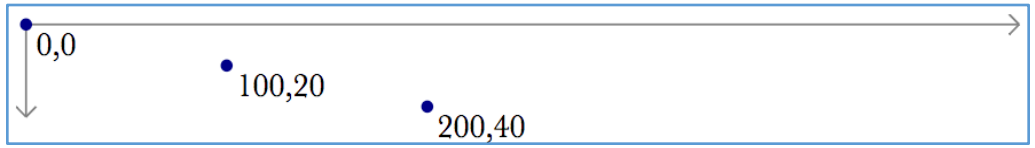
top
alignedlef
content
80-d
3.h
d3
tutorials

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>D3 Demo: 5-div bar chart</title>
        <script type="text/javascript" src="../d3/d3.v3.min.js"></script>
        <style type="text/css">

            div.bar {
                display: inline-block;
                width: 20px;
                height: 75px;   /* Gets overriden by D3-assigned height below */
                margin-right: 2px;
                background-color: teal;
            }

        </style>
    </head>
    <body>
        <script type="text/javascript">

            var dataset = [ 5, 10, 15, 20, 25 ];

            d3.select("body").selectAll("div")
                .data(dataset)
                .enter()
                .append("div")
                .attr("class", "bar")
                .style("height", function(d) {
                    var barHeight = d * 5;
                    return barHeight + "px";
                });

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

Power of Data

- More on data reading and generations: http://alignedleft.com/tutorials/d3/the-power-of-data

SVG simple shapes




```
<rect x="0" y="0" width="500" height="50"/>
```

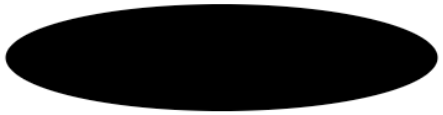
```
<line x1="0" y1="0" x2="500" y2="50" stroke="black"/>
```

```
<circle cx="250" cy="25" r="25"/>
```

```
<text x="250" y="25" font-family="sans-serif"
 font-size="25" fill="gray">Easy-peasy</text>
```

Easy-peasy

```
<ellipse cx="250" cy="25" rx="100" ry="25"/>
```



SVG Styles

- `fill` — A color value. Just as with CSS, colors can be specified as
  - named colors — `orange`
  - hex values — `#3388aa` or `#38a`
  - RGB values — `rgb(10, 150, 20)`
  - RGB with alpha transparency — `rgba(10, 150, 20, 0.5)`
- `stroke` — A color value.
- `stroke-width` — A numeric measurement (typically in pixels).
- `opacity` — A numeric value between 0.0 (completely transparent) and 1.0 (completely opaque).

With `text`, you can also use these properties,

- `font-family`
- `font-size`

SVG Examples

```
<circle cx="25" cy="25" r="22"
 fill="yellow" stroke="orange" stroke-width="5"/>
```

```
<circle cx="25" cy="25" r="22" class="pumpkin"/>
```

+

```
.pumpkin {
    fill: yellow;
    stroke: orange;
    stroke-width: 5;
 }
```

SVG Examples

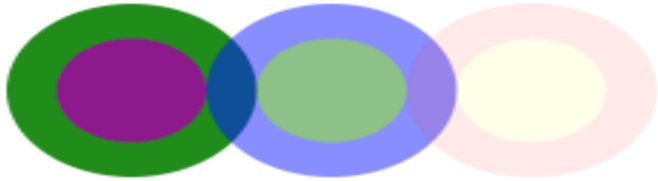
```
<circle cx="25" cy="25" r="20" fill="rgba(128, 0, 128, 1.0)"/>
<circle cx="50" cy="25" r="20" fill="rgba(0, 0, 255, 0.75)"/>
<circle cx="75" cy="25" r="20" fill="rgba(0, 255, 0, 0.5)"/>
<circle cx="100" cy="25" r="20" fill="rgba(255, 255, 0, 0.25)"/>
<circle cx="125" cy="25" r="20" fill="rgba(255, 0, 0, 0.1)"/>
```

SVG Examples

```
<circle cx="25" cy="25" r="20"
        fill="rgba(128, 0, 128, 0.75)"
        stroke="rgba(0, 255, 0, 0.25)" stroke-width="10"/>
<circle cx="75" cy="25" r="20"
        fill="rgba(0, 255, 0, 0.75)"
        stroke="rgba(0, 0, 255, 0.25)" stroke-width="10"/>
<circle cx="125" cy="25" r="20"
        fill="rgba(255, 255, 0, 0.75)"
        stroke="rgba(255, 0, 0, 0.25)" stroke-width="10"/>
```

SVG Examples

```
<circle cx="25" cy="25" r="20" fill="purple"
        stroke="green" stroke-width="10"
        opacity="0.9"/>
<circle cx="65" cy="25" r="20" fill="green"
        stroke="blue" stroke-width="10"
        opacity="0.5"/>
<circle cx="105" cy="25" r="20" fill="yellow"
        stroke="red" stroke-width="10"
        opacity="0.1"/>
```



Create SVG

```
var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 50);
```

Create SVG

Not Secure | alignedleft.com/content/03-tutorials/01-d3/110-drawing-svgs/3.html

Apps Gmail Banner Blackboard Rowan Email Rowan Google Drive Yahoo Mail Google Trans

Elements Console Sources Network Performance Memo

3.html

top
alignedlef
content
tutorials

```
11      <body>
12          <script type="text/javascript">
13
14              //Width and height
15              var w = 500;
16              var h = 50;
17
18              //Data
19              var dataset = [ 5, 10, 15, 20, 25 ];
20
21              //Create SVG element
22              var svg = d3.select("body")
23                          .append("svg")
24                          .attr("width", 500)
25                          .attr("height", 50);
26
27              var circles = svg.selectAll("circle")
28                  .data(dataset)
29                  .enter()
30                  .append("circle");
31
32              circles.attr("cx", function(d, i) {
33                          return (i * 50) + 25;
34                      })
35                  .attr("cy", h/2)
36                  .attr("r", function(d) {
37                          return d;
38                  });
39
40          </script>
41      </body>
42  </html>
```

Create SVG

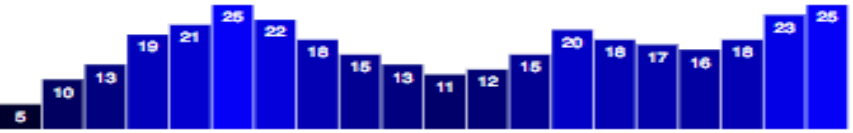
Elements Console Sources Network Performance Memory Application Secu

4.html ×

top
alignedlef
content

```
10      </head>
11      <body>
12          <script type="text/javascript">
13
14              //Width and height
15              var w = 500;
16              var h = 50;
17
18              //Data
19              var dataset = [ 5, 10, 15, 20, 25 ];
20
21              //Create SVG element
22              var svg = d3.select("body")
23                          .append("svg")
24                          .attr("width", w)
25                          .attr("height", h);
26
27              var circles = svg.selectAll("circle")
28                  .data(dataset)
29                  .enter()
30                  .append("circle");
31
32              circles.attr("cx", function(d, i) {
33                          return (i * 50) + 25;
34                      })
35                      .attr("cy", h/2)
36                      .attr("r", function(d) {
37                          return d;
38                      })
39                      .attr("fill", "yellow")
40                      .attr("stroke", "orange")
41                      .attr("stroke-width", function(d) {
42                          return d/2;
43                      });
44
45          </script>
46      </body>
47  </html>
```

Bar Graph using SVG



Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits

Page » | 10.html ×

- top
  - alignedleft.com
    - content/03-tut

```
4           <meta charset="utf-8">
5           <title>D3 Demo: Making a bar chart with value labels!</title>
6           <script type="text/javascript" src="../d3/d3.v3.min.js"></script>
7           <style type="text/css">
8               /* No style rules here yet */
9           </style>
10      </head>
11      <body>
12          <script type="text/javascript">
13
14              //Width and height
15              var w = 500;
16              var h = 100;
17              var barPadding = 1;
18
19              var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,
20                              11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];
21
22              //Create SVG element
23              var svg = d3.select("body")
24                          .append("svg")
25                          .attr("width", w)
26                          .attr("height", h);
27
28              svg.selectAll("rect")
29                 .data(dataset)
30                 .enter()
31                 .append("rect")
32                 .attr("x", function(d, i) {
33                      return i * (w / dataset.length);
34                 })
35                 .attr("y", function(d) {
36                      return h - (d * 4);
37                 })
38                 .attr("width", w / dataset.length - barPadding)
39                 .attr("height", function(d) {
40                      return d * 4;
41                 })
42                 .attr("fill", function(d) {
43                      return "rgb(0, 0, " + (d * 10) + ")";
44                 });
45
46              svg.selectAll("text")
47                 .data(dataset)
48                 .enter()
49                 .append("text")
50                 .text(function(d) {
51                      return d;
52                 })
53                 .attr("text-anchor", "middle")
54                 .attr("x", function(d, i) {
55                      return i * (w / dataset.length) + (w / dataset.length - barPadding) / 2;
56                 })
57                 .attr("y", function(d) {
58                      return h - (d * 4) + 14;
59                 })
60                 .attr("font-family", "sans-serif")
61                 .attr("font-size", "11px")
62                 .attr("fill", "white");
63
64
65
66          </script>
67      </body>
```

Scatter Plot

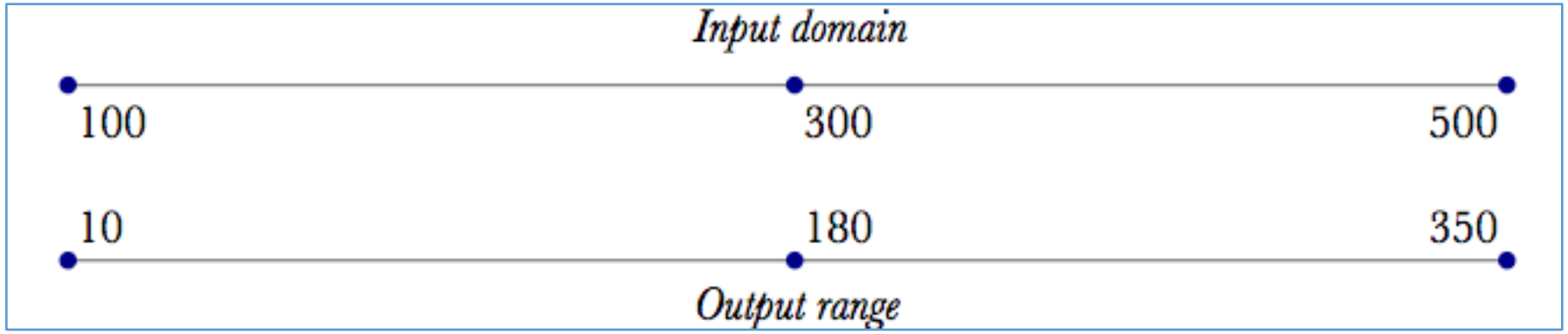
Elements Console Sources Network Performance Memory Application Security Audits

Page 3.html

top
alignedleft.com
content/03-tut
tutorials/d3/ma

```
10      </head>
11      <body>
12          <script type="text/javascript">
13
14              //Width and height
15              var w = 500;
16              var h = 100;
17
18              var dataset = [
19                              [5, 20], [480, 90], [250, 50], [100, 33], [330, 95],
20                              [410, 12], [475, 44], [25, 67], [85, 21], [220, 88]
21                            ];
22
23              //Create SVG element
24              var svg = d3.select("body")
25                          .append("svg")
26                          .attr("width", w)
27                          .attr("height", h);
28
29              svg.selectAll("circle")
30                 .data(dataset)
31                 .enter()
32                 .append("circle")
33                 .attr("cx", function(d) {
34                      return d[0];
35                 })
36                 .attr("cy", function(d) {
37                      return d[1];
38                 })
39                 .attr("r", function(d) {
40                      return Math.sqrt(h - d[1]);
41                 });
42
43              svg.selectAll("text")
44                 .data(dataset)
45                 .enter()
46                 .append("text")
47                 .text(function(d) {
48                      return d[0] + "," + d[1];
49                 })
50                 .attr("x", function(d) {
51                      return d[0];
52                 })
53                 .attr("y", function(d) {
54                      return d[1];
55                 })
56                 .attr("font-family", "sans-serif")
57                 .attr("font-size", "11px")
58                 .attr("fill", "red");
59
60          </script>
61      </body>
62  </html>
```

How to scale your data




```
var scale = d3.scale.linear()
                    .domain([100, 500])
                    .range([10, 350]);
```

```
scale(100);   //Returns 10
scale(300);   //Returns 180
scale(500);   //Returns 350
```

More at http://alignedleft.com/tutorials/d3/scales

Axes

- http://alignedleft.com/content/03-tutorials/01-d3/160-axes/4.html

Transitions

- CSS3 transitions with D3 are *magical*!
- D3 interpolates values for you…

```
rect.attr("height", 0)
rect.transition( )
        .delay( 500 ) //can be a function of data
        .duration(200) //can be a function of data
        .attr("height", 5) //can be a function of data
        .style("fill","green") //can be a function of data
```

So transitions allow a vis to be dynamic…

But they're not really interactive…

.on( )

```
rect.on ("click", function(d){
   d.color = "blue";
   redraw( rawdata )
})
```

d is the data point backing the element clicked on

HTML Events

- click
- mouseover
- mouseenter
- mouseout
- etc.

E-U-E Pattern Template

```
var group = vis.selectAll("rect")
      .data(rawdata) //rawdata must be an array!
group.enter( ).append("rect") //ENTER!
      .attr( )
      .style( )
group //UPDATE!
      .attr( )
      .style( )
group.exit( ).remove( ) //EXIT!
```

Many online examples