

# D3.js

UC Berkeley Graduate School of Journalism

Review

# Review — CSS selectors

IDs are unique, don't reuse the name you give

```
<div id="title"></div>
```

Classes are not, you can use them  
to refer to multiple elements

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

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

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

# Review — CSS selectors

To refer to an ID, use the pound sign **#title**

```
<div id="title"></div>
```

To refer to all classes, use the dot **.listing**

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

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

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

# Review — DataTypes

```
324230293420
```

**Number**

```
"Hello World"
```

**String**

```
false
```

**Boolean**

```
[3, 4, 3, 2, 1, 0]
```

**Array**

```
{ hair:"black" }
```

**Object Literal**

```
function(){}
```

**Function**

# Review — Calling Functions

This is a function that returns a string:

```
function(){ return "Hey";}
```

This is also calling a function, sending data:

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

```
something("yo") //returned yo
```

# Review — Calling Functions

So, if I wrote:

```
function(d){ return d + 5;}
```

```
something(10) //returns 15  
something(5)  //returns 10  
something(0)  //returns 5
```

# Review — SVG

SVG container element

```
<svg width="400" height="300">
```

```
</svg>
```

# Review — SVG

add circle element to appear on screen.

```
<svg width="400" height="300">
```

```
<circle cx="100" cy="100" r="10" />
```

```
</svg>
```

# Review — SVG

or a rectangle

```
<svg width="400" height="300">
```

```
<circle cx="100" cy="100" r="10" />
```

```
<rect x="10" y="10" width="10" ..
```

```
</svg>
```

# Review — SVG transform

```
<g transform="">  
</g>
```

```
<g transform="">  
</g>
```

```
<g transform="">  
</g>
```

# Review — SVG transform

```
<g transform="translate(30, 50)">  
</g>
```

```
<g transform="">  
</g>
```

```
<g transform="">  
</g>
```

# Review — SVG transform

```
<g transform="translate(30, 50)">  
</g>
```

```
<g transform="rotate(-40, 10, 10)">  
</g>
```

```
<g transform="">  
</g>
```

# Review — SVG transform

```
<g transform="translate(30, 50)">  
</g>
```

```
<g transform="rotate(-40, 10, 10)">  
</g>
```

```
<g transform="scale(1.5)">  
</g>
```

# Review — SVG transform

or combine them

```
<g transform="translate(30, 50)rotate(-40, 10, 10)scale(1.5)">  
</g>
```

**D3**

# d3js.org

You need to include the library before  
you can use the commands.

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

# D3

# d3

this is a single variable, with the whole d3 library saved inside of it as an object. You can access the various tools through **dot notation**.

## d3 — Selecting elements on the webpage

```
<div id="something">  
</div>
```

# d3 — Selecting elements on the webpage

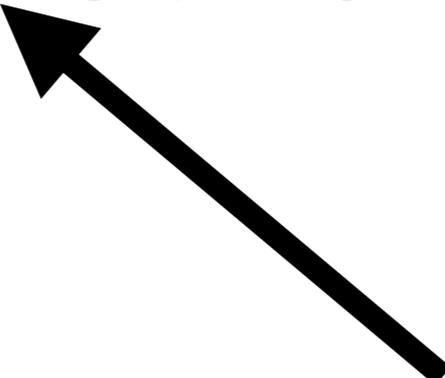
```
<div id="something">  
</div>
```

**d3**

## d3 — Selecting elements on the webpage

```
<div id="something">  
</div>
```

**d3.select("#something")**



## d3 — Selecting elements on the webpage

```
<div id="something">  
  <p></p>  
</div>
```

```
d3.select("#something")  
  .append("p")
```

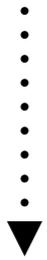
## d3 — Selecting elements on the webpage

```
<div id="something">  
  <p>Hello World</p>  
</div>
```

```
d3.select("#something")  
  .append("p")  
  .text("Hello World")
```

# D3 Select

always start with d3 variable



```
d3.select("#title")
```

# D3 Select

select is the first command to select the element



```
d3.select("title")
```

# D3 Select

a string which contains the CSS selector of the element



```
d3.select("#title")
```

# D3 Select

dot means it's a class



```
d3.select(".headlines")
```

This will only return the first  
element with `class="headlines"`

# D3 Select

```
d3.selectAll(".headlines")
```

This will return an array, containing  
ALL of the elements with  
`class="headlines"`

# D3 Recap

selects id="title"

**.select("#title")**

selects everything with class="headlines"

**.selectAll(".headlines")**

appends an element, like a div

**.append("div")**

places text within currently selected element

**.text("Here is text")**

# D3 Recap

sets an attribute on the currently selected tag

```
.attr("width", 300)
```

set some CSS styles on the currently selected tag

```
.style({"fill": "red"})
```

just like text, but renders html

```
.html("<p>hey</p>")
```

adds or removes a class attribute

```
.classed("headlines", false)
```

# Chaining

# D3 Chaining

<code>d3.select("#title")</code>	◀ returns #title
<code>  .append("p")</code>	◀ appends p, returns p
<code>  .text("Here is text")</code>	◀ adds text to p

<code>d3.select("#title")</code>	◀ returns #title
<code>  .text("Here is text")</code>	◀ adds text to #title
<code>  .append("p")</code>	◀ appends p

# D3 Chaining

```
d3.select("#title")  
  .append("p")  
  .text("Here is text")
```

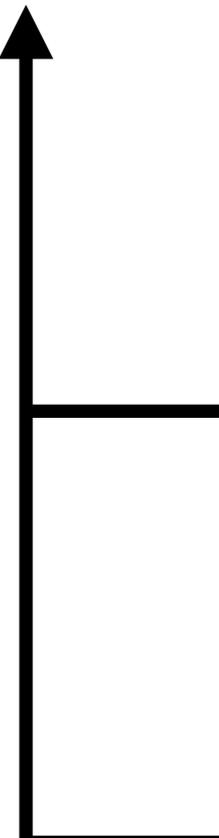
<div id="title">  
 <p>Here is text</p>  
</div>

```
d3.select("#title")  
  .text("Here is text")  
  .append("p")
```

<div id="title">  
 Here is text  
 <p></p>  
</div>

# D3 Chaining

```
var title = d3.select("#title")  
              .append("div");  
  
title.append("p")  
      .text("hey there");  
  
title.append("p")  
      .text("Another p tag");
```



# D3 Chaining

```
var title = d3.select("#title")  
              .append("div");
```

```
title.append("p")  
      .text("hey there");
```

```
title.append("p")  
      .text("Another p tag");
```

# Drawing SVGs

# We'll create this SVG with D3

1. we need svg



```
<svg width="500" height="400">
```

```
  <circle cx="250" cy="200" r="30" />
```

```
</svg>
```

# We'll create this SVG with D3

2. we need width height attributes



```
<svg width="500" height="400">
```

```
<circle cx="250" cy="200" r="30" />
```

```
</svg>
```

# We'll create this SVG with D3

```
<svg width="500" height="400">
```

```
  <circle cx="250" cy="200" r="30" />
```

```
</svg>
```

3. we need to append a circle

# We'll create this SVG with D3

```
<svg width="500" height="400">
```

```
  <circle cx="250" cy="200" r="30" />
```

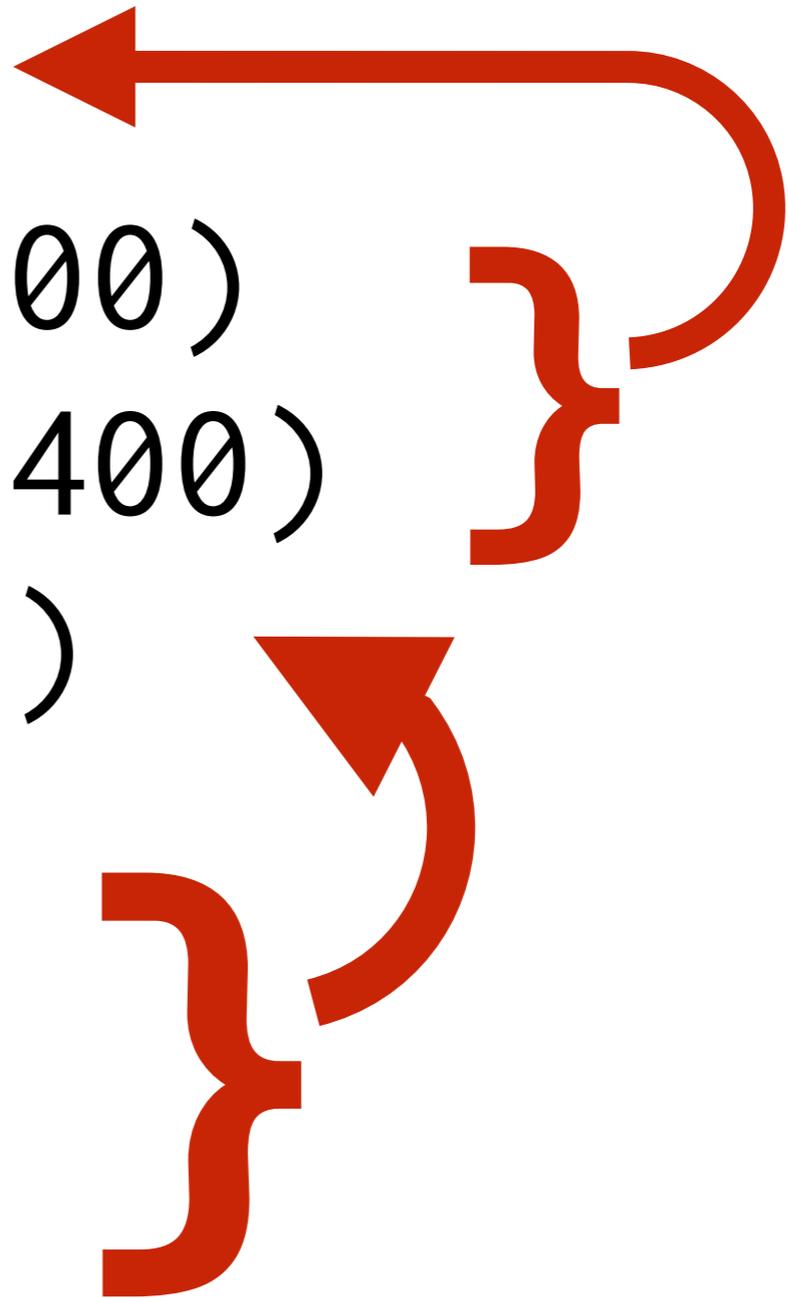
```
</svg>
```



4. we need to add attributes for circle

# D3 to draw SVG

```
d3.select("body")  
  .append("svg")  
    .attr("width", 500)  
    .attr("height", 400)  
  .append("circle")  
    .attr("cx", 250)  
    .attr("cy", 200)  
    .attr("r", 30)
```



# Using Data with D3

# D3 wants an Array of data

this is an array of numbers

```
[39, 239, 12, 32, 42, 24]
```

or an array of objects

```
[  
  {name: "Apples", value: 39},  
  {name: "Orange", value: 23},  
  {name: "Pears", value: 30},  
  {name: "Bananas", value: 40}  
]
```

# Data in D3

```
var svg = d3.select("body")
          .append("svg")
          .attr("width", 400)
          .attr("height", 300);
```



this variable now has the SVG

# Data in D3

```
svg.selectAll(".bubbles")
```



there is NO .bubbles

# Data in D3

```
svg.selectAll(".bubbles")  
  .data([32, 43, 23, 54])  
  .enter() ◀ ..... Magic!
```

# Data in D3

```
svg.selectAll(".bubbles")  
  .data([32, 43, 23, 54])  
  .enter()↑↑↑↑
```

code that is in this red box

# Data in D3

```
svg.selectAll(".bubbles")  
  .data([32, 43, 23, 54])  
  .enter()  
  .append("circle")
```

# Data in D3

```
svg.selectAll(".bubbles")  
  .data([32, 43, 23, 54])  
  .enter()  
  .append("circle")  
  .attr("cy", 100)
```

# Data in D3

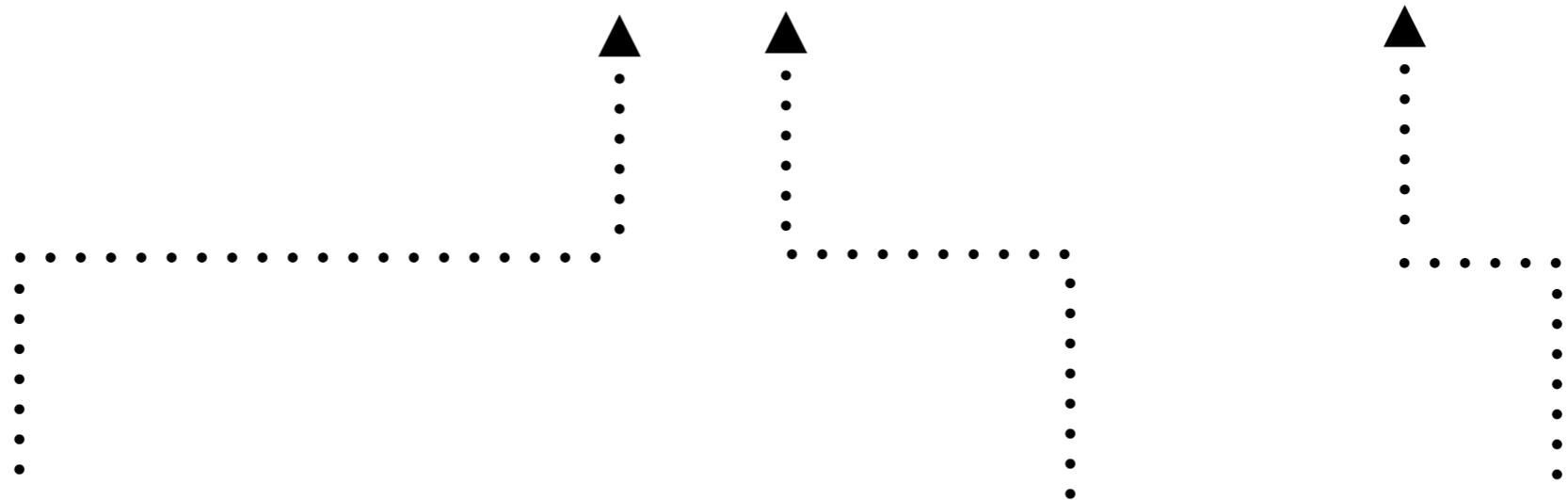
```
svg.selectAll(".bubbles")
  .data([32, 43, 23, 54])
  .enter()
  .append("circle")
  .attr("cy", 100)
  .attr("cx", function(d, i){
    return i * 50;
  })
```

# Data in D3

```
svg.selectAll(".bubbles")
  .data([32, 43, 23, 54])
  .enter()
  .append("circle")
  .attr("cy", 100)
  .attr("cx", function(d, i){
    return i * 50;
  })
  .attr("r", function(d, i){
    return d;
  })
```

# Data in D3

```
.attr("r", function(d, i){ return d; })
```



**d** is the value of the data array during this iteration of the loop.

**i** is the index (how many times) this loop has run.

**return** is what you want to return