

Lab 6

D3.js (Part I)

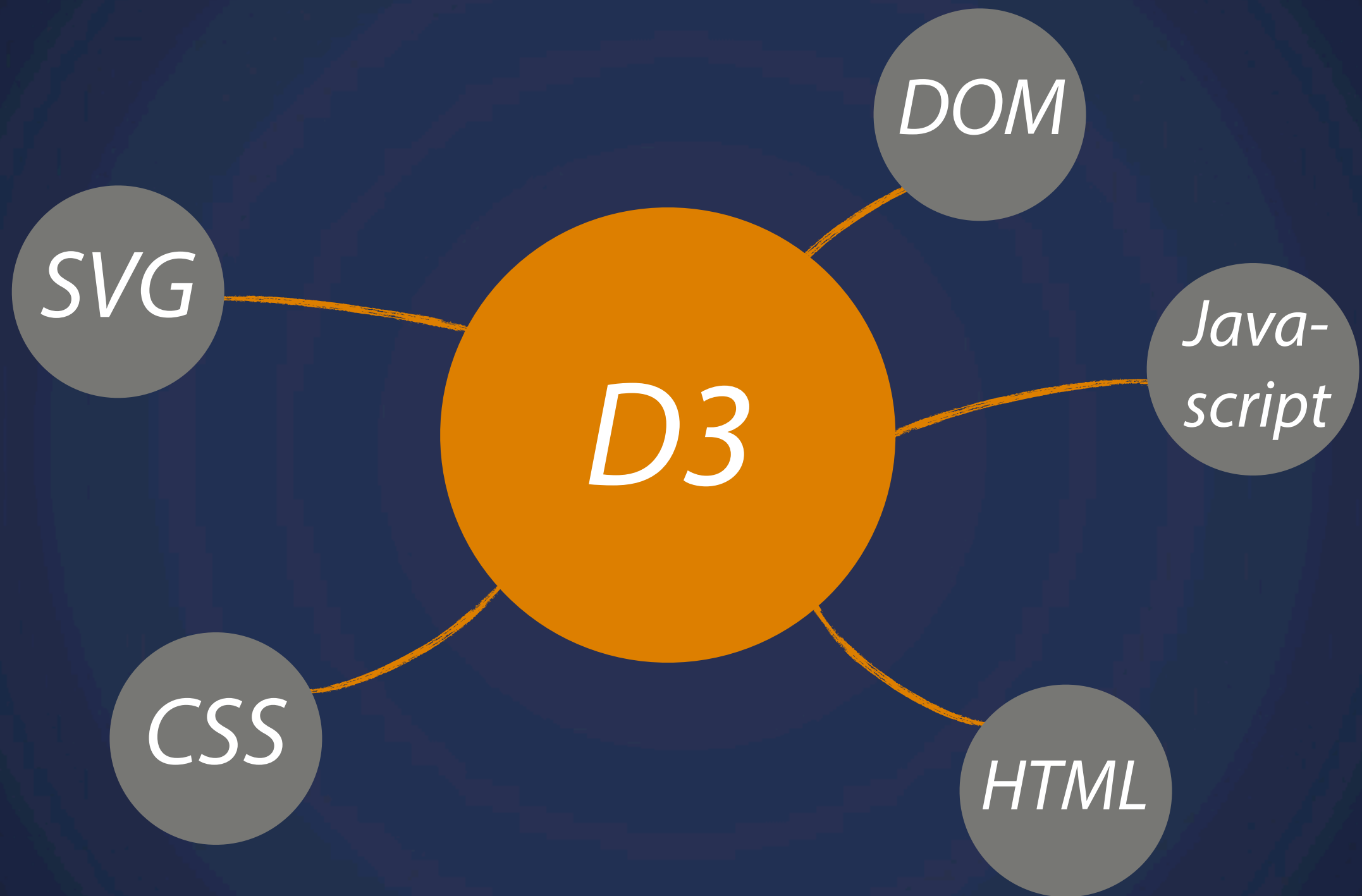
February 28, 2013 – Michael Porath (@poezn)

What is D3?

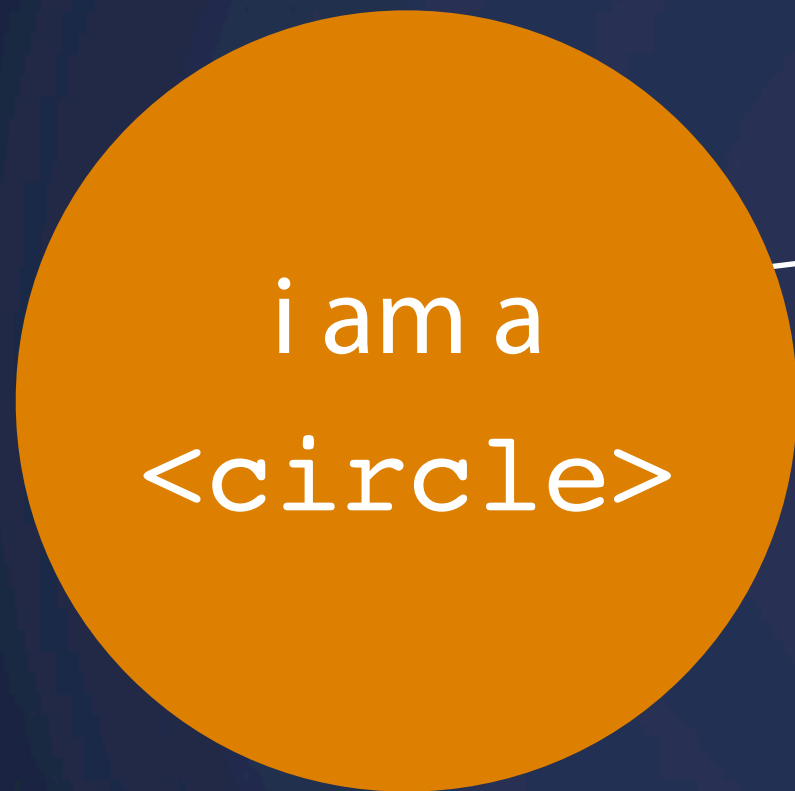
“D3.js is a JavaScript library for manipulating documents based on data.”

D3 = Data-
Driven
Documents

However...

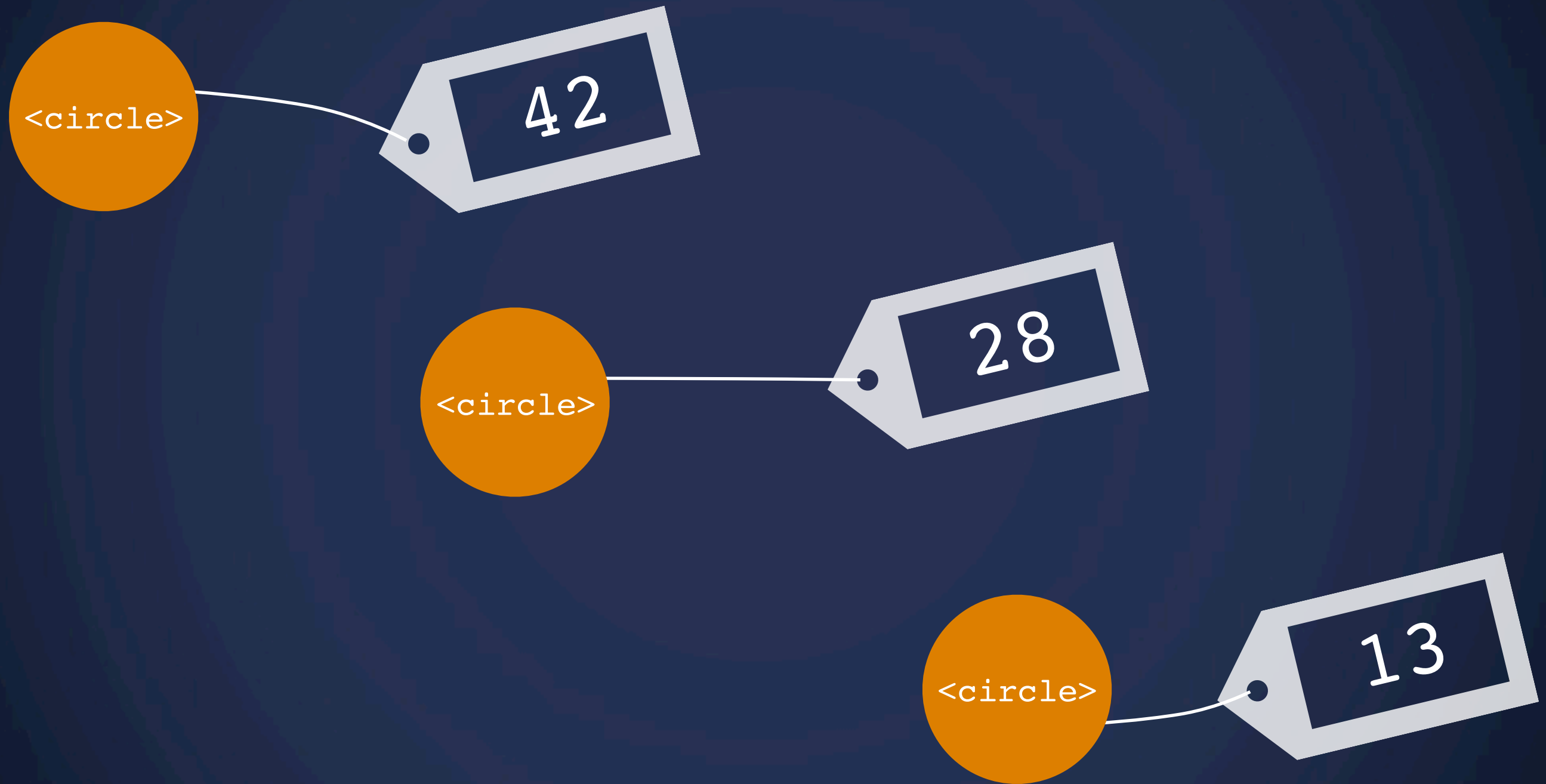


Data Binding



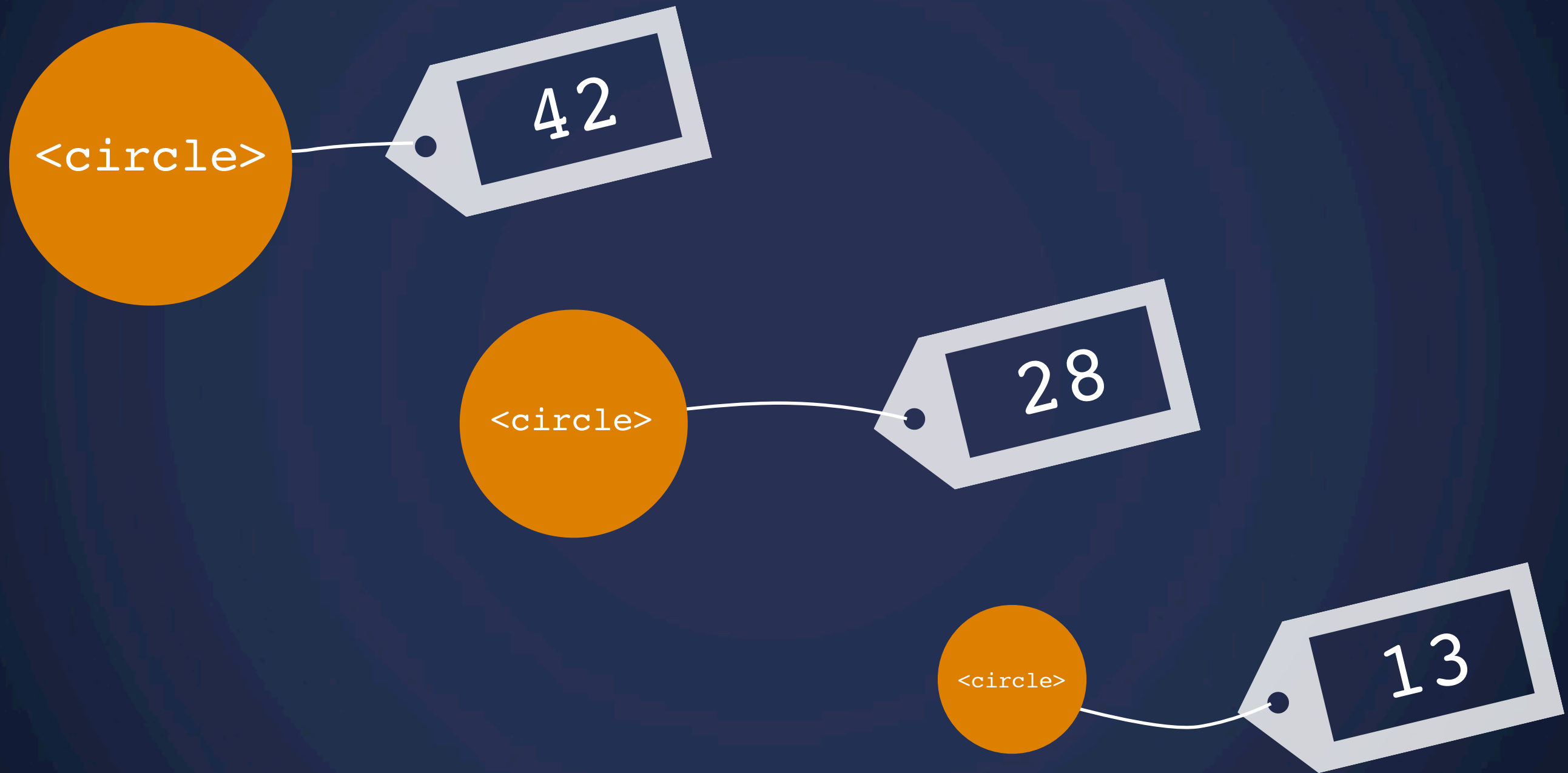
each `<element>` has
a datum "attached"

Data Binding

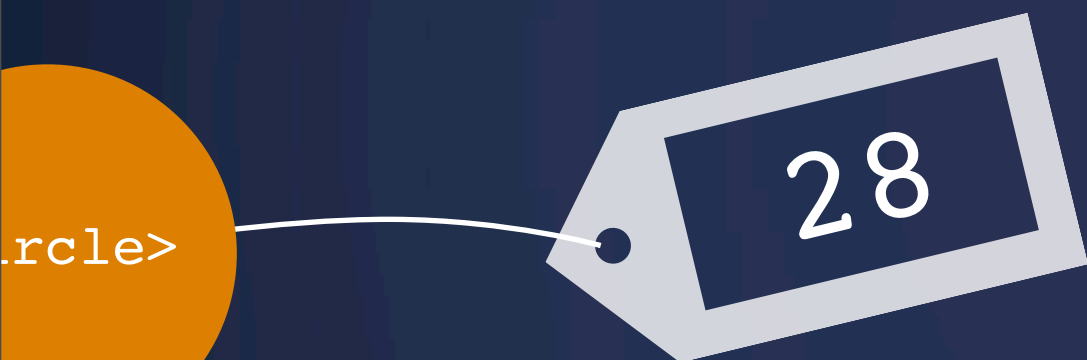
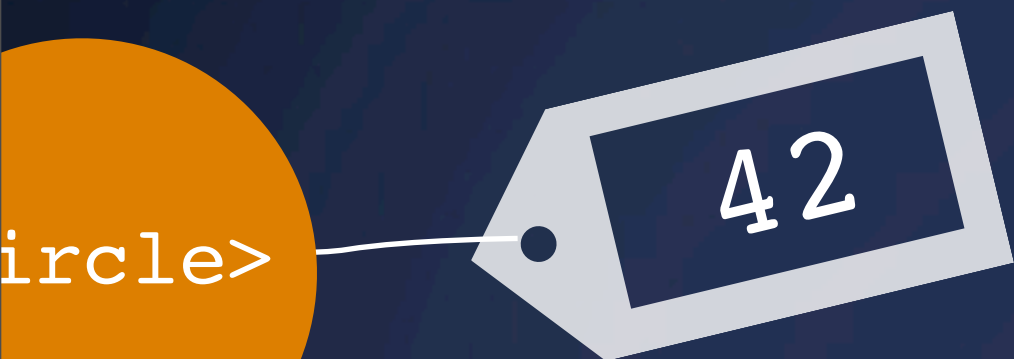


Data Binding

Data Driven Attributes. Here: radius



And now in D3



```
d3.selectAll("circle")  
  .data([42, 28, 13])  
  .enter().append("circle")  
  .attr({  
    "r": function(d) {  
      return d;  
    }  
  });
```

Step By Step

Selection

```
d3.selectAll("circle")
```

Data Binding

```
.data([42, 28, 13])
```

enter / exit

```
.enter().append("circle")
```

Setting attributes
and styles

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


Step By Step

Select a number of <elements> from the DOM

Selection

Data Binding

enter / exit

Setting attributes
and styles

```
d3.selectAll("circle")  
  .data([42, 28, 13])  
  .enter().append("circle")  
  .attr({  
    "r": function(d) {  
      return d;  
    }  
  });
```

Step By Step

Assign a datum to each previously selected elements

Selection

Data Binding

enter / exit

Setting attributes
and styles

```
d3.selectAll("circle")  
  .data([42, 28, 13])  
  .enter().append("circle")  
  .attr({  
    "r": function(d) {  
      return d;  
    }  
  });
```

Step By Step

The number of selected elements don't have to match up with the number of data points. How do we deal with that?

Selection

```
d3.selectAll("circle")
```

```
.data([42, 28, 13])
```

Data Binding

```
.enter().append("circle")
```

```
.attr({
```

```
  "r": function(d) {
```

```
    return d;
```

```
  }
```

```
});
```

enter / exit

Setting attributes
and styles

Step By Step

*How should our elements look like and behave?
Here is where the data gets important!*

Selection

```
d3.selectAll("circle")
```

```
.data([42, 28, 13])
```

```
.enter().append("circle")
```

```
.attr({
```

```
  "r": function(d) {
```

```
    return d;
```

```
  }
```

```
});
```

Data Binding

enter / exit

Setting attributes
and styles

Step By Step

Data binding and enter/exit statements are not necessary if you work with existing elements

Selection

```
d3.selectAll("circle")
```

```
.data([42, 28, 13])
```

```
.enter().append("circle")
```

```
.attr({
```

```
  "r": function(d) {
```

```
    return d;
```

```
  }
```

```
});
```

Data Binding

enter / exit

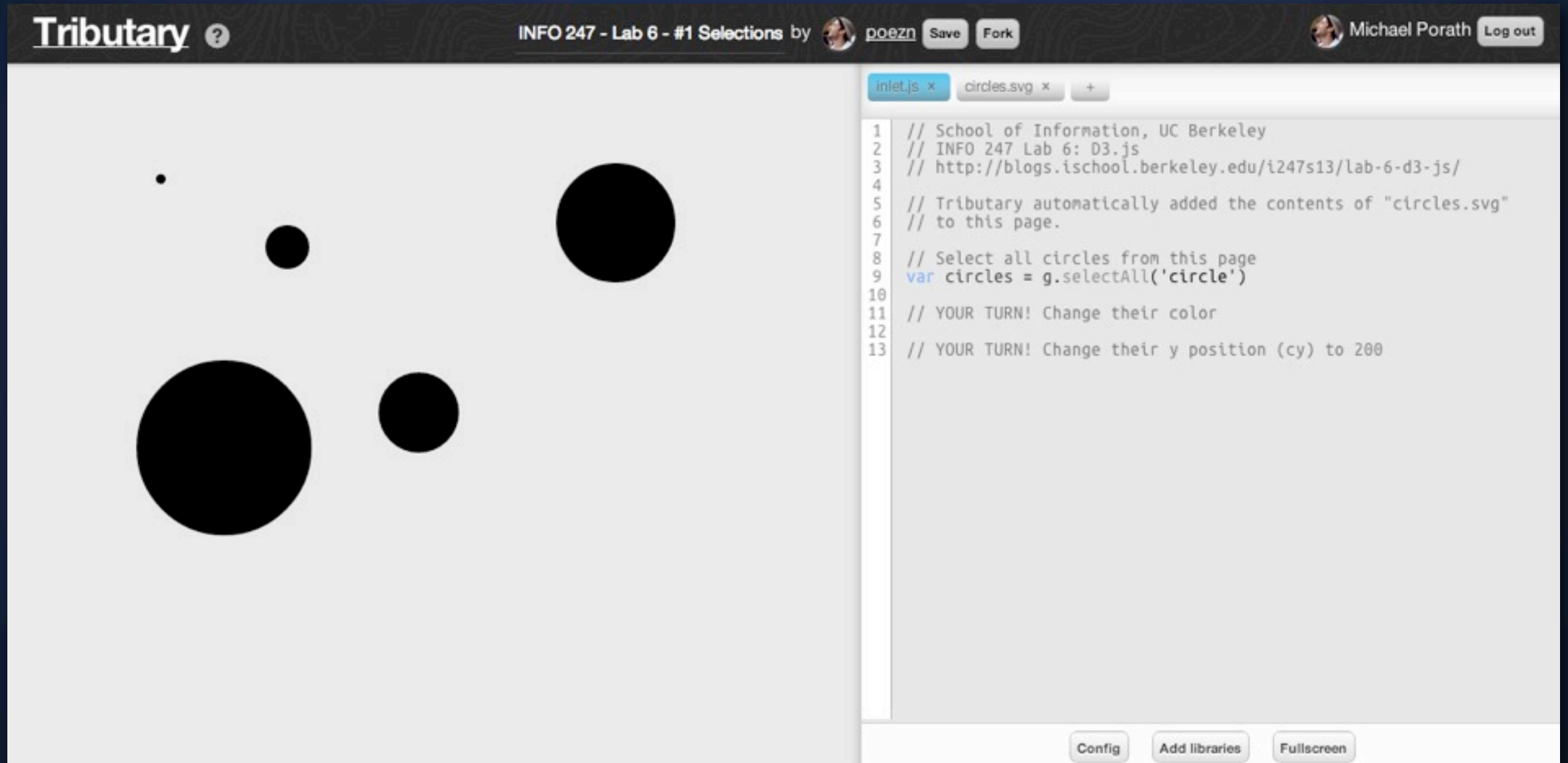
Setting attributes
and styles

Your Turn

Let's try this

Selections

Exercise 1



The screenshot shows a Tributary.io workspace. The top navigation bar includes the Tributary logo, the project name "INFO 247 - Lab 6 - #1 Selections" by user "poezn", and a user profile for "Michael Porath" with a "Log out" button. The workspace is split into two main areas: a canvas on the left and a code editor on the right. The canvas displays five black circles of different sizes and positions. The code editor shows the following JavaScript code:

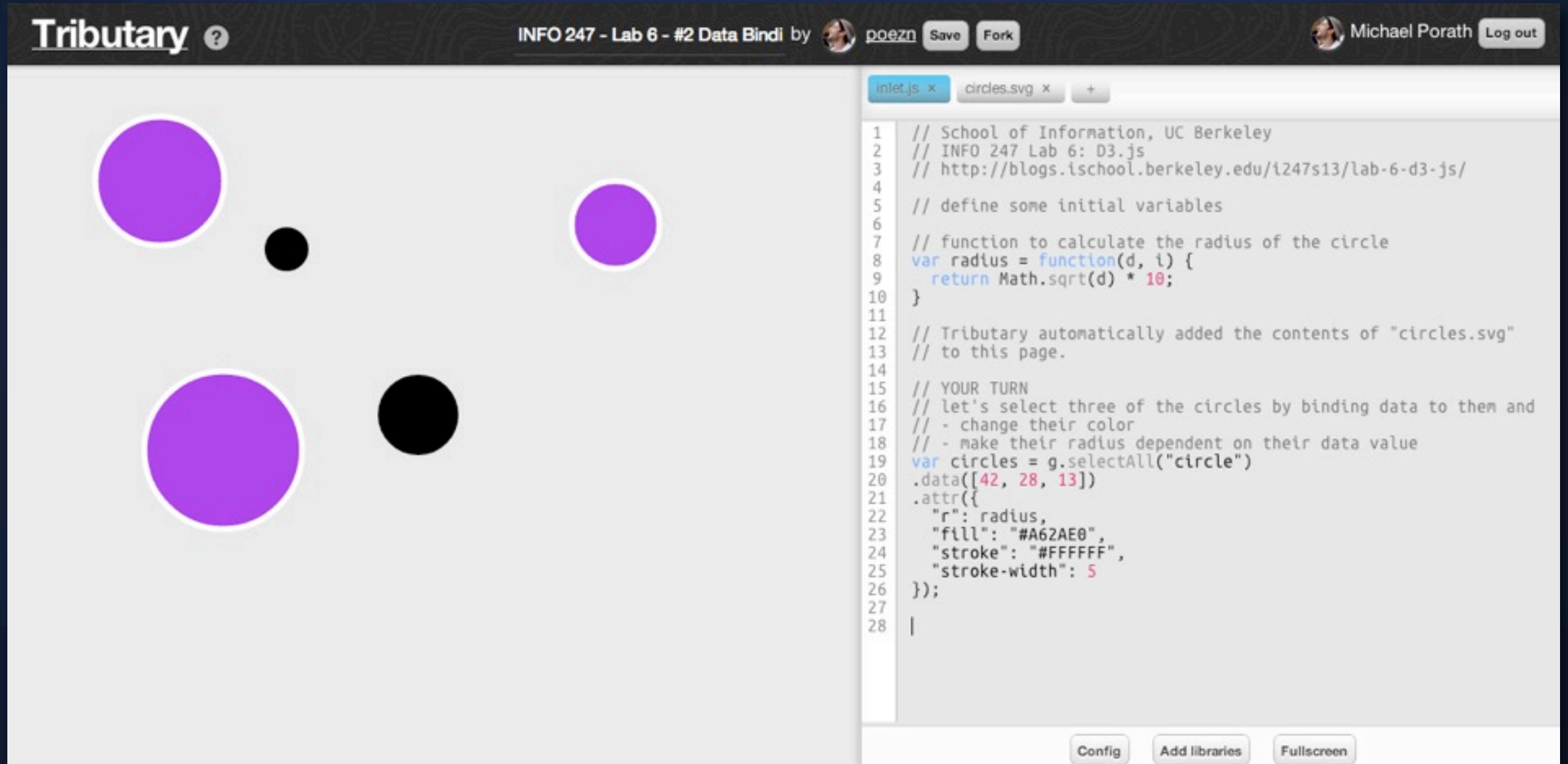
```
1 // School of Information, UC Berkeley
2 // INFO 247 Lab 6: D3.js
3 // http://blogs.ischool.berkeley.edu/i247s13/lab-6-d3-js/
4
5 // Tributary automatically added the contents of "circles.svg"
6 // to this page.
7
8 // Select all circles from this page
9 var circles = g.selectAll('circle')
10
11 // YOUR TURN! Change their color
12
13 // YOUR TURN! Change their y position (cy) to 200
```

At the bottom of the code editor, there are three buttons: "Config", "Add libraries", and "Fullscreen".

<http://tributary.io/inlet/5058838>

Selections

Exercise 2



The screenshot shows a Tributary.io workspace. On the left, a visualization displays five circles on a light gray background. Two circles are purple with a white stroke, and three are black with a white stroke. The circles vary in size, with the largest purple circle at the bottom left and the smallest black circle in the top middle. On the right, a code editor shows the following JavaScript code:

```
1 // School of Information, UC Berkeley
2 // INFO 247 Lab 6: D3.js
3 // http://blogs.ischool.berkeley.edu/~t247s13/lab-6-d3-js/
4
5 // define some initial variables
6
7 // function to calculate the radius of the circle
8 var radius = function(d, i) {
9   return Math.sqrt(d) * 10;
10 }
11
12 // Tributary automatically added the contents of "circles.svg"
13 // to this page.
14
15 // YOUR TURN
16 // let's select three of the circles by binding data to them and
17 // - change their color
18 // - make their radius dependent on their data value
19 var circles = g.selectAll("circle")
20 .data([42, 28, 13])
21 .attr({
22   "r": radius,
23   "fill": "#A62AE0",
24   "stroke": "#FFFFFF",
25   "stroke-width": 5
26 });
27
28 |
```

At the bottom of the code editor, there are three buttons: "Config", "Add libraries", and "Fullscreen".

<http://tributary.io/inlet/5059349>

Enter / Exit

Exercise 3

```
d3.selectAll("circle")
```

all existing circles



Enter / Exit

Exercise 3

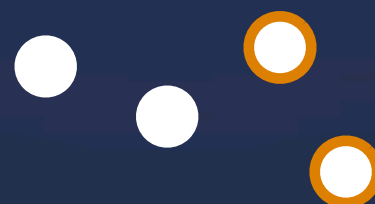
```
d3.selectAll("circle")
```

```
d3.selectAll("circle")  
  .data([1, 2, 3, 4])  
  .enter()  
  .append("circle")
```

all existing circles



*creates circles with new
data bindings*



Enter / Exit

Exercise 3

```
d3.selectAll("circle")
```

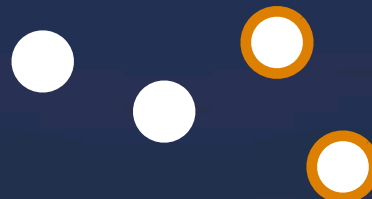
```
d3.selectAll("circle")  
  .data([1, 2, 3, 4])  
  .enter()  
  .append("circle")
```

```
d3.selectAll("circle")  
  .data([1, 2, 3])  
  .exit()  
  .remove()
```

all existing circles



creates circles with new data bindings

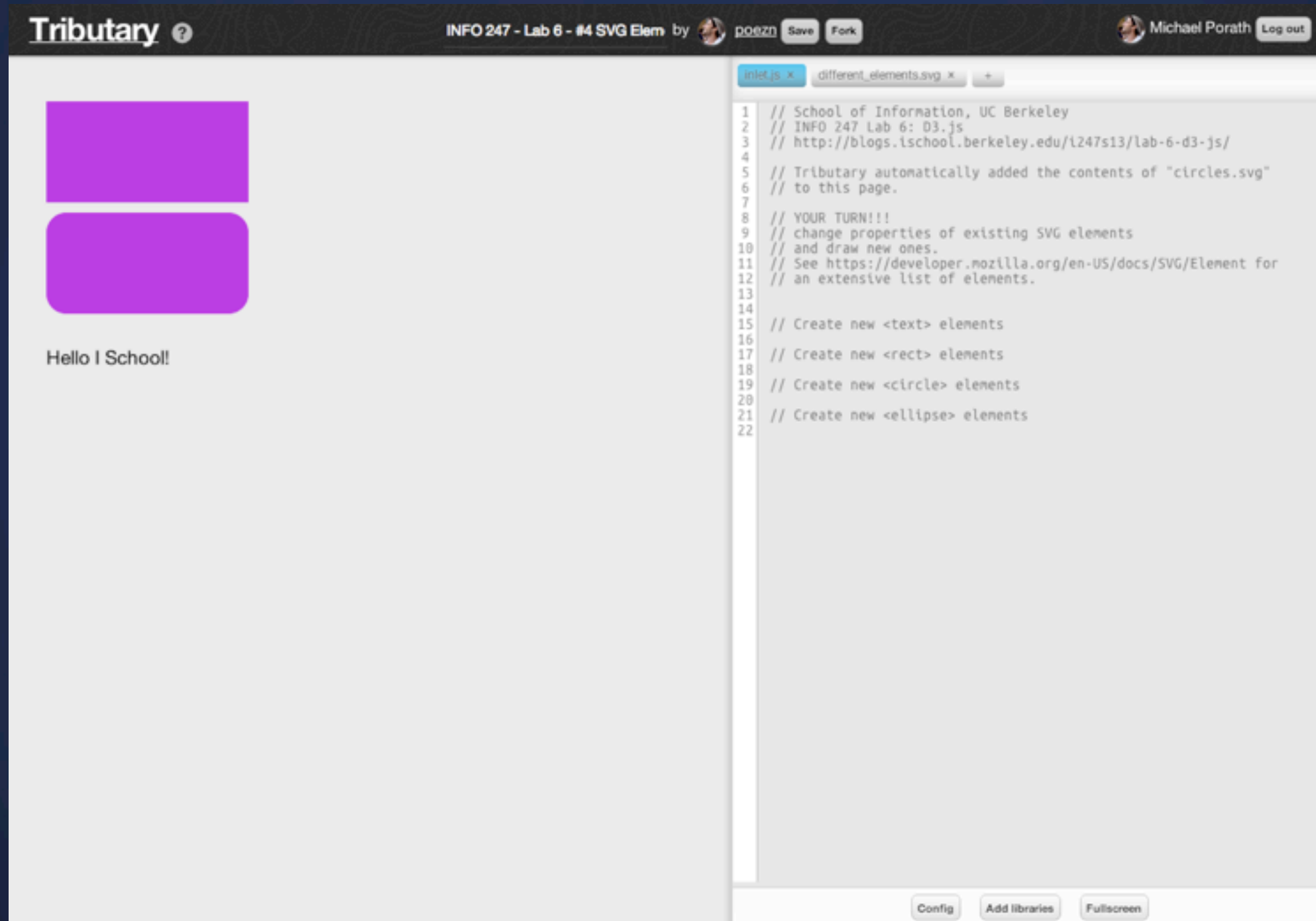


removes circles without data bindings



Different SVG elements

Exercise 4



The screenshot shows a web browser window with a Tributary page. The page has a dark header with the Tributary logo, the title "INFO 247 - Lab 6 - #4 SVG Elem" by "doezn", and a user profile for "Michael Porath" with a "Log out" button. The main content area displays two purple shapes: a rectangle and a rounded rectangle. Below the shapes is the text "Hello I School!". On the right side, there is a code editor with two tabs: "inlet.js" and "different_elements.svg". The code in the "different_elements.svg" tab is as follows:

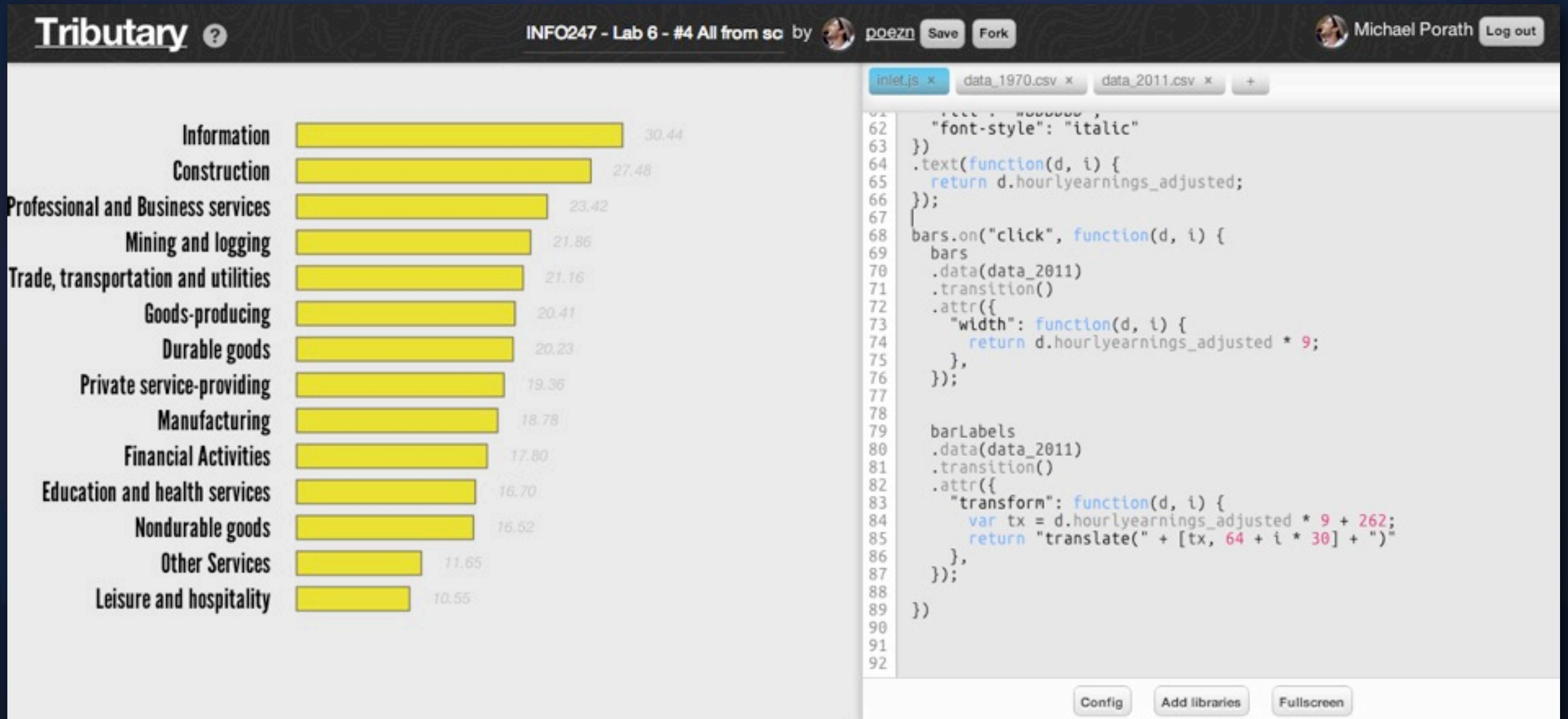
```
1 // School of Information, UC Berkeley
2 // INFO 247 Lab 6: D3.js
3 // http://blogs.ischool.berkeley.edu/t247s13/lab-6-d3-js/
4
5 // Tributary automatically added the contents of "circles.svg"
6 // to this page.
7
8 // YOUR TURN!!!
9 // change properties of existing SVG elements
10 // and draw new ones.
11 // See https://developer.mozilla.org/en-US/docs/SVG/Element for
12 // an extensive list of elements.
13
14
15 // Create new <text> elements
16 // Create new <rect> elements
17 // Create new <circle> elements
18 // Create new <ellipse> elements
19
20
21
22
```

At the bottom of the code editor, there are three buttons: "Config", "Add libraries", and "Fullscreen".

<http://tributary.io/inlet/5059912>

All together

Exercise 5 (only if you're bored!)



<http://tributary.io/inlet/5059603>

What next?

Practice, practice, practice!

great set of D3.js tutorials
for beginners!

<http://alignedleft.com/tutorials/>

Next Lecture

Interaction

Next Lab

D3.js (Part II)