

D3 Exercises

Slides adapted from...

Maneesh Agrawala

Jessica Hullman

Ludwig Schubert

Peter Washington

Alec Glassford and Zach Maurer

Gracie Young and Vera Lin

I. Run a local server

Create a file named `index.html` with the text “Hello, world!”

Display it in the browser

I. Run a local server

```
python -m http.server
```

```
python -m SimpleHTTPServer
```

Type “localhost:<port_number>” in a web browser. You should see a webpage displaying “Hello, world!”

II. Draw a blue circle

Create a file named `index.html` with the text “Hello, world!”

Display it in the browser

II. Draw a blue circle

```
<script>
```

```
  var svg = d3.select('#vis');  
  var circle = svg.append('circle');
```

```
  circle.attr('r', 10)  
          .attr('cx', 100)  
          .attr('cy', 100)  
          .style('fill', 'blue');
```

```
</script>
```

III. Draw circles according to data

```
var data = [  
  { x : 1, y : 2 },  
  { x : 2, y : 1 },  
  { x : 3, y : 4 },  
  { x : 4, y : 5 },  
  { x : 5, y : 3 }  
];
```

III. Draw circles according to data

```
svg.selectAll('myCircles')  
  .data(data)  
  .enter().append('circle')  
    .attr('r', 10)  
    .style('fill', 'blue')  
    .attr('cx', function(d) { return d.x; })  
    .attr('cy', function(d) { return d.y; });
```

IV. Make data easier to read with scales

```
d3.scaleLinear()
```


IV. Make data easier to read with scales

```
x = d3.scaleLinear()
  .domain([0,5])
  .range([10,400]);
y = d3.scaleLinear()
  .domain([0,5])
  .range([400,10]);

svg.selectAll('circle')
  .data(data)
  .enter().append('circle')
    .attr('r', 10)
    .style('fill', 'blue')
    .attr('cx', function(d) { return x(d.x); })
    .attr('cy', function(d) { return y(d.y); });
```

V. Make circles draggable

use: `d3.drag()`

will need to check out:

<https://github.com/d3/d3-drag>

V. Make circles draggable

```
svg.selectAll('circle')  
  .data(data)  
  .enter().append('circle')  
    .attr('r', 10)  
    .style('fill', 'blue')  
    .attr('cx', function(d) { return x(d.x); })  
    .attr('cy', function(d) { return y(d.y); })  
    .call(d3.drag().on('drag', callback));
```

```
function callback(d) {  
  d3.select(this)  
    .attr('cx', d.x = d3.event.x)  
    .attr('cy', d.y = d3.event.y);  
}
```

VI. Make green circle behind blue circle

One possible implementation strategy:

- 1) Add an id variable to all of the data.
- 2) Use this id when naming the id attribute of the circles.
- 3) In the drag function for the inner circle, drag the corresponding outer circle (based on id)

VI. Make green circle behind blue circle

```
var circles = svg.selectAll('circle')
    .data(data) // bind data to circles

circle.enter().append('circle') // add green circles (behind)
    .attr('r', 20)
    .style('fill', 'green')
    .attr('cx', function(d) { return xscale(d.x); })
    .attr('cy', function(d) { return yscale(d.y); })
    .attr('id', function(d, i) { return 'id' + i.toString(); })

circles.enter().append('circle') // add blue circles (in front)
    .attr('r', 10)
    .style('fill', 'blue')
    .attr('cx', function(d) { return xscale(d.x); })
    .attr('cy', function(d) { return yscale(d.y); })
    .attr('id', function(d, i) { return 'id' + i.toString(); })
    .call(d3.drag().on('drag', function(d, i) {
        d3.selectAll('#id' + i.toString()) // select both circles
            .attr('cx', d.x = d3.event.x) by id
            .attr('cy', d.y = d3.event.y);
    }));
```

VII. Make circles with $x > 3$ disappear on button click

VII. Make circles with $x > 3$ disappear on button click

1) Create button

2) Handle click

3) Filter data

4) Update circles

VII. Make circles with $x > 3$ disappear on button click

```
drawCircles(data);
```

```
function drawCircles(inputData) {  
  var circles = svg.selectAll('circle')  
    .data(inputData);  
  circles.enter().append('circle')  
    .attr('r', 10)  
    .style('fill', 'blue')  
    .attr('cx', function(d) { return xscale(d.x); })  
    .attr('cy', function(d) { return yscale(d.y); })  
  circles.exit().remove();  
}
```

```
d3.select('#button')  
  .on('click', function() {  
    var newdata = data.filter(d => (+d.x <= 3));  
    drawCircles(newdata);  
  });
```


VII. Make circles with $x > 3$ disappear on button click

```
drawCircles(data); // draw initial circles

function drawCircles(inputData) {
  var circles = svg.selectAll('circle')
    .data(inputData);
  circles.enter().append('circle') // add circles @ inputData
    .attr('r', 10)
    .style('fill', 'blue')
    .attr('cx', function(d) { return xscale(d.x); })
    .attr('cy', function(d) { return yscale(d.y); })
  circles.exit().remove(); // remove old circles
}

d3.select('#button') // button has id '#button'
  .on('click', function() { // click handler
    var newdata = data.filter(d => (+d.x <= 3)); // filter
    drawCircles(newdata); // redraw circles
  });
```