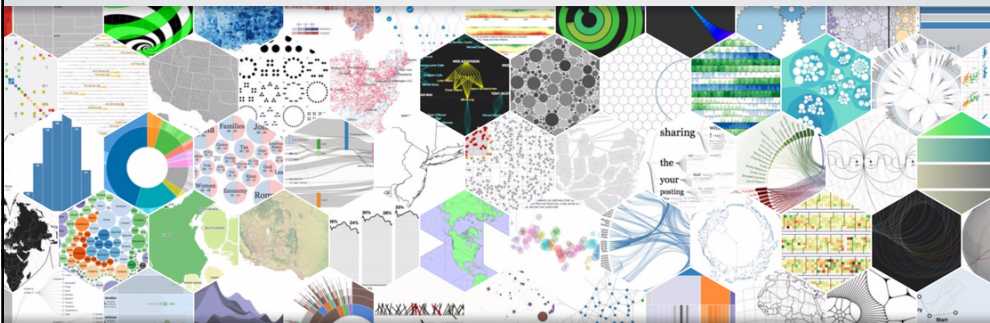




DATA-DRIVEN DOCUMENTS

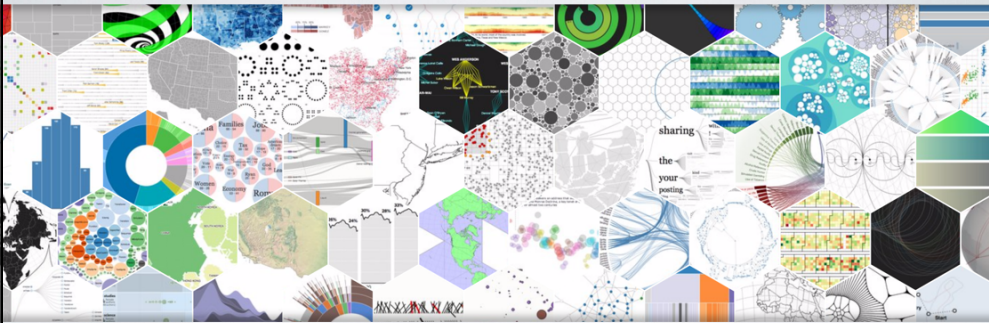
(d3.js, D3, D³)

It's just a toolkit...



Georgia Tech Visualization Lab

It's just a toolkit...



a really powerful toolkit, but you still have to do all of the programming and design.

Fall 2017 3 CS 4460

Georgia Tech Visualization Lab

WHAT DOES D3 DO?

- A Javascript framework for generating, styling and animating web content based on data.
- *What does D3 do?*
 - **Load** data into the browser's memory
 - **Create, Update, and Remove** web page elements based on data (referred to as "binding data to elements on the web page")
 - **Transform and Style** those page elements based on data
 - **Transition** elements between states in response to user input

Fall 2017 4 CS 4460

WHAT DOES D3 DO?

- D3 provides a large offering of data-to-visual mappings for you to use in **transforming and styling** your data visualization

Axes	Geographies	Scales
Brushes	Hierarchies	Shapes
Chords	Interpolators	Time Formats
Colors	Number Formats	Time Intervals
Dragging	Paths	Transitions
Easings	Polygons	Voronoi Diagrams
Forces	Quadrees	Zooming

- While D3 provides a framework to apply these mappings, it is up to you to design how to use these rules in your visualization.

D3.JS - ORIGINS

- D3 was originally written by Mike Bostock, Vadim Ogievetsky, and Jeff Heer
- Bostock started working on D3 as a PhD Student at Stanford
- D3 Predecessors (Jeff Heer's work):
 - ProtoVis & Prefuse - Java toolkit
 - Flare - ActionScript that runs on Adobe Flash Player
- D3's success stems from:
 - piggy-backing on web technologies
 - thorough documentation
 - growing community of programmers
 - Visit <https://bl.ocks.org/> - the D3 community showcase

The Gold Standard in Web Visualization

- D3.js appears everywhere on the web:
 - From data journalism pieces at the *New York Times*, *Wall Street Journal*, and *Washington Post* to...
 - The blogs and data visualization groups scattered about the web (e.g. <http://pudding.cool/> and <http://bocoup.com/> are worth a look)
- D3 certainly owes some of its popularity to riding the *everything-on-the-web trend*
- Nevertheless, the declarative way in which you can express relationships between data and visual elements is *fundamentally superior to any other library!*

D3.JS - AN EXAMPLE



D3.JS - AN EXAMPLE

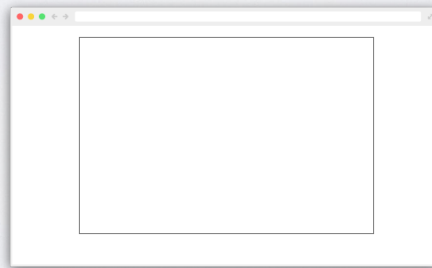
```

<script>
  var numericData = [1, 2, 4, 8, 16];

  var svg = d3.select('svg');

  // Add rectangles
  svg.selectAll('rect')
    .data(numericData)
    .enter()
    .append('rect');
</script>

```



D3.JS - AN EXAMPLE

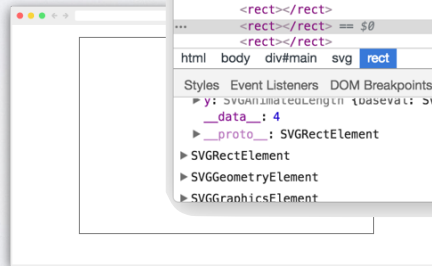
```

<script>
  var numericData = [1, 2, 4, 8, 16];

  var svg = d3.select('svg');

  // Add rectangles
  svg.selectAll('rect')
    .data(numericData)
    .enter()
    .append('rect');
</script>

```



Developer Tools - http://localhost:7071/

Elements Memory Sources Network Audits >> 1

```

<!DOCTYPE html>
<html lang="en">
<head>...</head>
<body>
  <div id="main">
    <svg width="300" height="200" style="border: 1px solid #777;">
      <rect></rect>
      <rect></rect>
      <rect></rect>
      <rect></rect>
      <rect></rect>
    </svg>
  </div>
</body>
</html>

```

html body div#main svg rect

Styles Event Listeners DOM Breakpoints Properties

```

y: SVGAnimatedLengthList baseVal: SVGLength animVal: SVGLength
__data__: 4
__proto__: SVGRectElement
SVGRectElement
SVGGeometryElement
SVGGraphicsElement

```

D3.JS - AN EXAMPLE

```

<script>
var numericData = [1, 2, 4, 8, 16];
var svg = d3.select('svg');

// Add rectangles
svg.selectAll('rect')
  .data(numericData)
  .enter()
  .append('rect')
  .attr('fill', '#f77e46')
  .attr('width', 30)
  .attr('height', 40);
</script>

```



Fall 2017

||

CS 4460

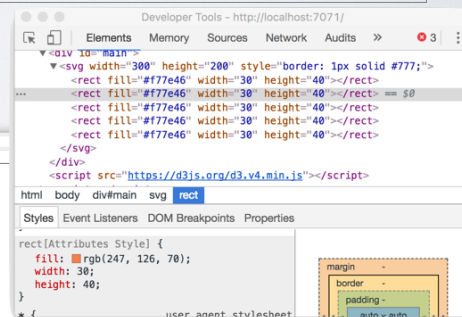
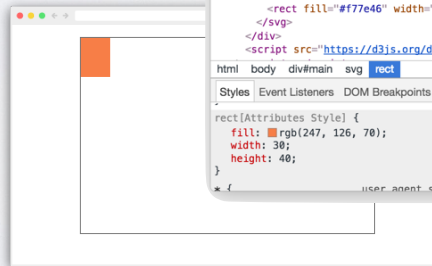
D3.JS - AN EXAMPLE

```

<script>
var numericData = [1, 2, 4, 8, 16];
var svg = d3.select('svg');

// Add rectangles
svg.selectAll('rect')
  .data(numericData)
  .enter()
  .append('rect')
  .attr('fill', '#f77e46')
  .attr('width', 30)
  .attr('height', 40);
</script>

```



Fall 2017

12

CS 4460

D3.JS - AN EXAMPLE

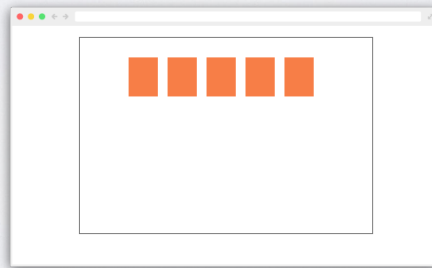
```

<script>
var numericData = [1, 2, 4, 8, 16];

var svg = d3.select('svg');

// Add rectangles
svg.selectAll('rect')
  .data(numericData)
  .enter()
  .append('rect')
  .attr('fill', '#f77e46')
  .attr('width', 30)
  .attr('height', 40)
  .attr('y', 20)
  .attr('x', function(d, i) {
    return (i * 40) + 50;
  });
</script>

```



D3.JS - AN EXAMPLE

```

<script>
var numericData = [1, 2, 4, 8, 16];

var svg = d3.select('svg');

// Add rectangles
svg.selectAll('rect')
  .data(numericData)
  .enter()
  .append('rect')
  .attr('fill', '#f77e46')
  .attr('width', 30)
  .attr('height', function(d) {
    return 160 * d / 16;
  })
  .attr('y', 20)
  .attr('x', function(d, i) {
    return (i * 40) + 50;
  });
</script>

```



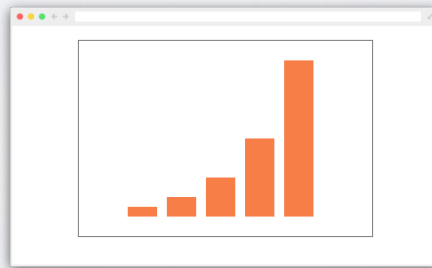
D3.JS - AN EXAMPLE

```

<script>
  var numericData = [1, 2, 4, 8, 16];
  var svg = d3.select('svg');

  // Add rectangles
  svg.selectAll('rect')
    .data(numericData)
    .enter()
    .append('rect')
    .attr('fill', '#f77e46')
    .attr('width', 30)
    .attr('height', function(d){
      return 160 * d / 16;
    })
    .attr('y', function(d){
      return 180 - 160 * d / 16;
    })
    .attr('x', function(d, i) {
      return (i * 40) + 50;
    });
</script>

```



LAB PROCEDURE

Before Class

- Read Chapters 5 and 6 - *Interactive Data Visualization for the Web* by Scott Murray
- Git pull example code (<https://github.gatech.edu/CS-4460/Labs.git>)

In-Class

- Open Lab 3 instruction page (<https://github.gatech.edu/CS-4460/Labs/wiki>)
- Work through activities
- ★ First thing, start a simple http server with python at 03_lab directory

After Class

- If you run out of time, finish all lab activities
- Lab Solutions will be posted Monday nights