Intro to d3.js: Relevant Links

Repo for our workshop:

http://bit.ly/2ofmTCB

D3 API Documentation:

https://github.com/d3/d3/blob/master/API.md

MDN's JavaScript Documentation:

https://developer.mozilla.org/en-US/docs/Web/JavaScript

Bl.ocks (the home for sharing d3 modules and examples):

https://bl.ocks.org/

SVG Documentation:

https://developer.mozilla.org/en-US/docs/Web/SVG

An example visualization:

https://ligo.northwestern.edu/media/mass-plot/index.html

ADVANCED D3:

Curve Interpolations: http://bit.ly/2xR9VxX

Voronoi: http://bit.ly/2gK35mi

https://bl.ocks.org/mbostock/4060366

Zooming: http://bit.ly/2f6WYsg

Brushing/Zooming: http://bit.ly/2lTjfef

Stacking: http://bl.ocks.org/mbostock/582915

Collision Constraint: http://bit.ly/2vOPk05

Dragging: http://bit.ly/2j6x5gH

"Each": https://bl.ocks.org/mbostock/9490313

Dispatching (super advanced): https://bl.ocks.org/mbostock/5872848

Modules/Plugins (*most* advanced): https://bost.ocks.org/mike/d3-plugin

"Store-bought" (and other) viz solutions:

Plotly: https://plot.ly/

ggplot2: http://ggplot2.org/

Matplotlib: https://matplotlib.org/

Tableau: https://www.tableau.com/

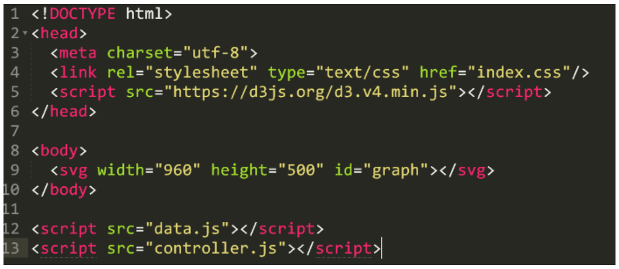
Processing: https://processing.org/

Intro to d3.js: Review Topics

Most vizzies are *box-mix* cake

D3 is a tiramisu

Steps to starting with d3.js:

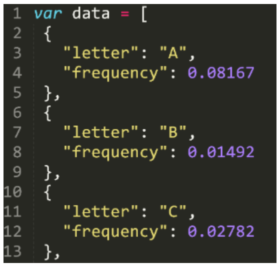


```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="index.css"/>
  <script src="https://d3js.org/d3.v4.min.js"></script>
</head>

<body>
  <svg width="960" height="500" id="graph"></svg>
</body>

<script src="data.js"></script>
<script src="controller.js"></script>
```

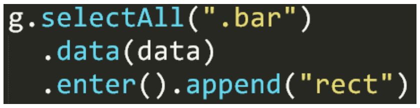
1. Build your environment:
    a. Set up your HTML
    b. Link your CSS file
    c. Link your libraries (d3 included)
    d. Link your main JavaScript
2. Examine your data:
    a. Is it in the right format? Structured correctly?
    b. Do you need a web server or can you host it locally?
3. Link your data to your document:
    a. Will you load it in using d3 or will you load it in up front?
4. Create your starting element:
    a. Use an id to select it easily, make sure it is an <svg> type



```
var data = [
 {
   "letter": "A",
   "frequency": 0.08167
 },
 {
   "letter": "B",
   "frequency": 0.01492
 },
 {
   "letter": "C",
   "frequency": 0.02782
 },
```

Steps to every basic visualization in d3.js:

1. **Create your scales**: these convert your data (domain) into the pixel space of the web (range)
2. **Select** an element, then "selectAll" on its children (even if you haven't made them yet)
3. Bind **data** to this selection and "enter" the data
4. **Append** an element (this appends one element per data piece):
    a. SVG types: rect, circle, text, etc
    b. Even HTML types work: div, p, span, etc
5. **Modify** styling, attributes, classes or interactivity to each element, based on the data
    a. .attr("width", function(d,i) { return x(d.frequency) }
6. Optional: Create an axis, title, interactive pieces, animations, etc



```
g.selectAll(".bar")
  .data(data)
  .enter().append("rect")
```