

D3.js Tutorial

(Hands on Session)

Ayush Kumar and Klaus Mueller

D3: Data-Driven Documents

D3 Show Reel

www.BANDICAM.com

- AAPL
- AMZN
- IBM
- MSFT

<https://bl.ocks.org/mbostock/1256572>

D3: Data-Driven Documents

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

Data
Driven
Transformation

D3: Technology & Concept

- D3 Library
- JQuery (Bonus)
- Data Visualization
- JSON/CSV
- Array & Objects
- SVG – Scalable Vector Graphics
- Transitions
- Data Scaling
- Data Binding
- Data Display & Charting

Directory Structure

Replace “**CSE564**” with your “**Project_Folder**” name

- ❑ CSE564/
 - ❑ index.html

- ❑ CSE564 /lib/
 - ❑ d3.v4.js

- ❑ CSE564 /js/
 - ❑ test.js

<http://www3.cs.stonybrook.edu/~aykumar/CSE564.zip>

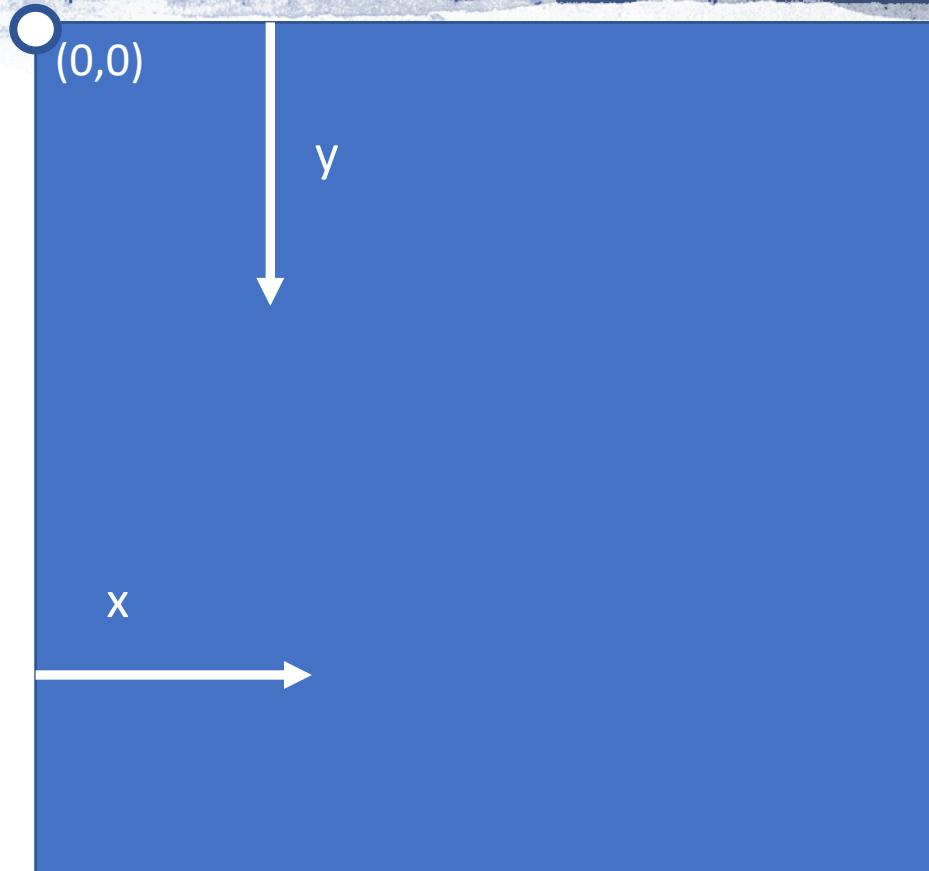
- ❑ CSE564 /css/
 - ❑ style.css

- ❑ CSE564 /data/
 - ❑ iris.csv

VERY VERY IMPORTANT !!!

Please do not be hesitant to ask questions!

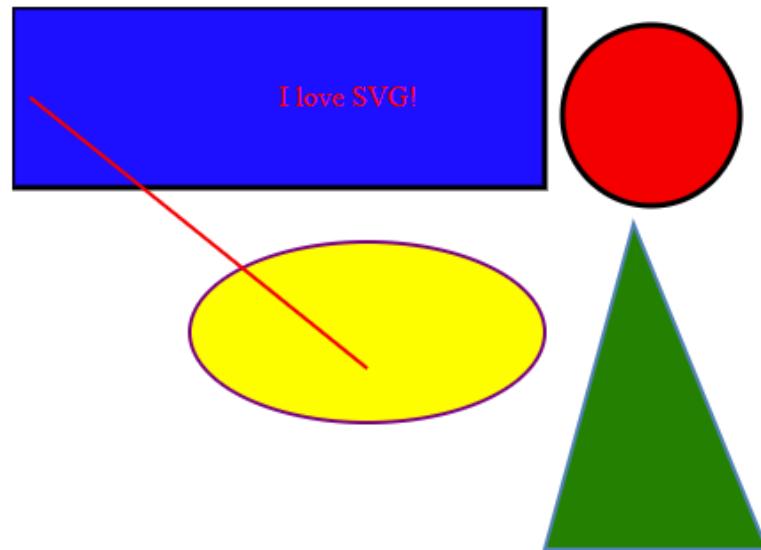
SVG Basics



- [SVG](#) is an [XML](#) language, similar to [XHTML](#), which can be used to draw vector graphics.
- It can be used to create an image either by specifying all the lines and shapes necessary, by modifying already existing raster images, or by a combination of both.

SVG Shapes

- <rect>
 - ```
<svg width="400" height="110">
 <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
</svg>
```
- <circle>
  - ```
<svg height="100" width="100">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```
- <ellipse>
 - ```
<svg height="140" width="500">
 <ellipse cx="200" cy="80" rx="100" ry="50" style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>
```
- <line>
  - ```
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-width:2" />
</svg>
```
- <text>
 - ```
<svg height="30" width="200">
 <text x="0" y="15" fill="red">I love SVG!</text>
</svg>
```
- <path>
  - ```
<svg height="500" width="500">
  <path d="M350 120 L300 300 L425 300 Z" style="fill:green;stroke:steelblue;stroke-width:2" />
</svg>
```



AND NOW D3 ...

- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION

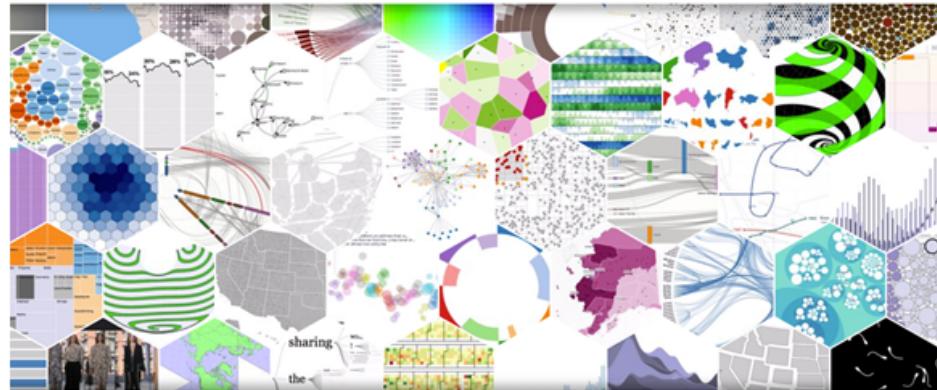
How to make this Work ???

D3 Begins...

Setup D3.js Development Environment



- D3 library
- Web server
- Editor
- Web browser



D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

[See more examples.](#)

Download the latest version (4.6.0) here:

• [d3.zip](#)

Download D3 source

To link directly to the latest release, copy this snippet:

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

Direct link

D3 Begins...

Setup D3.js Development Environment

- D3 library
- Web server
- Editor
- Web browser

```
<script src="../d3.js"></script>
```

```
<script src="../d3.min.js"></script>
```

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

D3 Begins...

Setup D3.js Development Environment

- D3 library
- Web server
- Editor
- Web browser

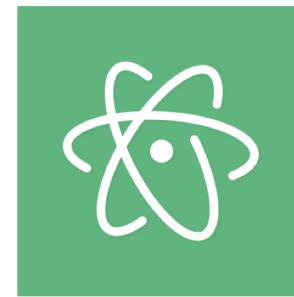
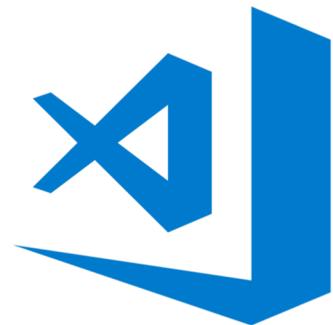
```
python -m SimpleHTTPServer 8000
```

<http://localhost:8000/>

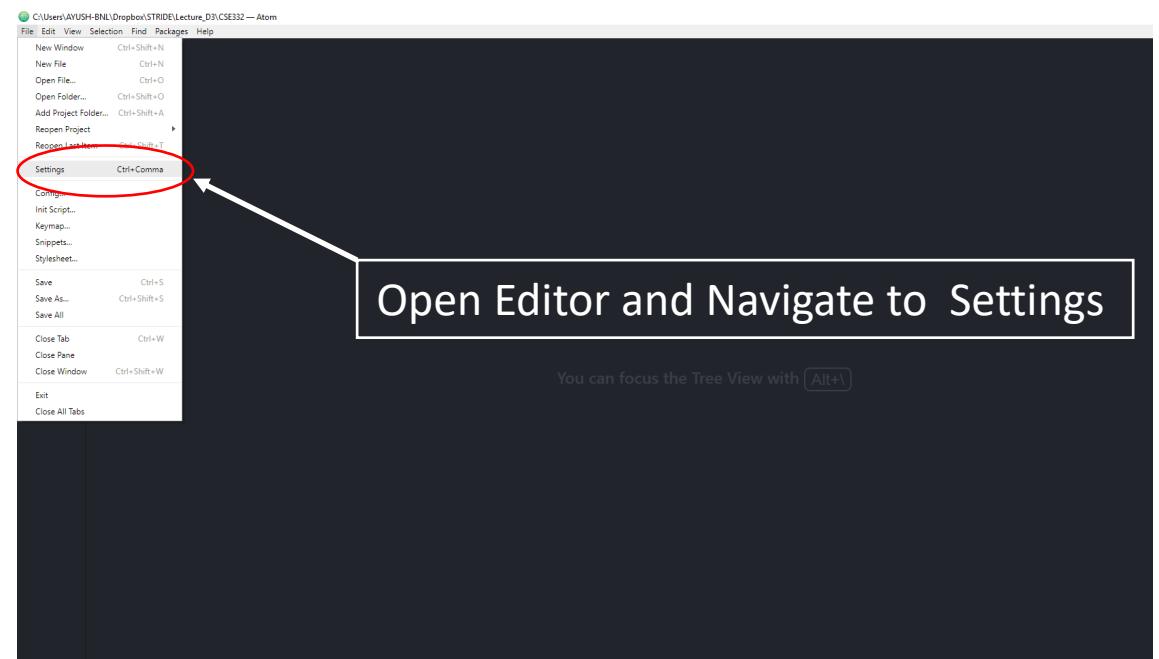
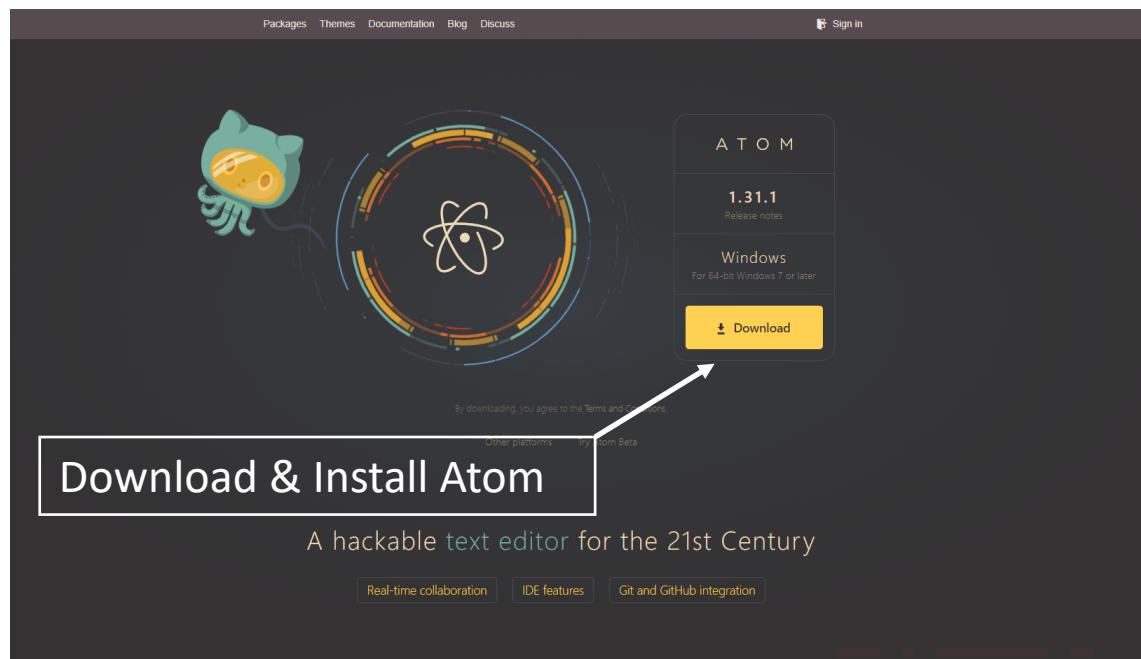
D3 Begins...

Setup D3.js Development Environment

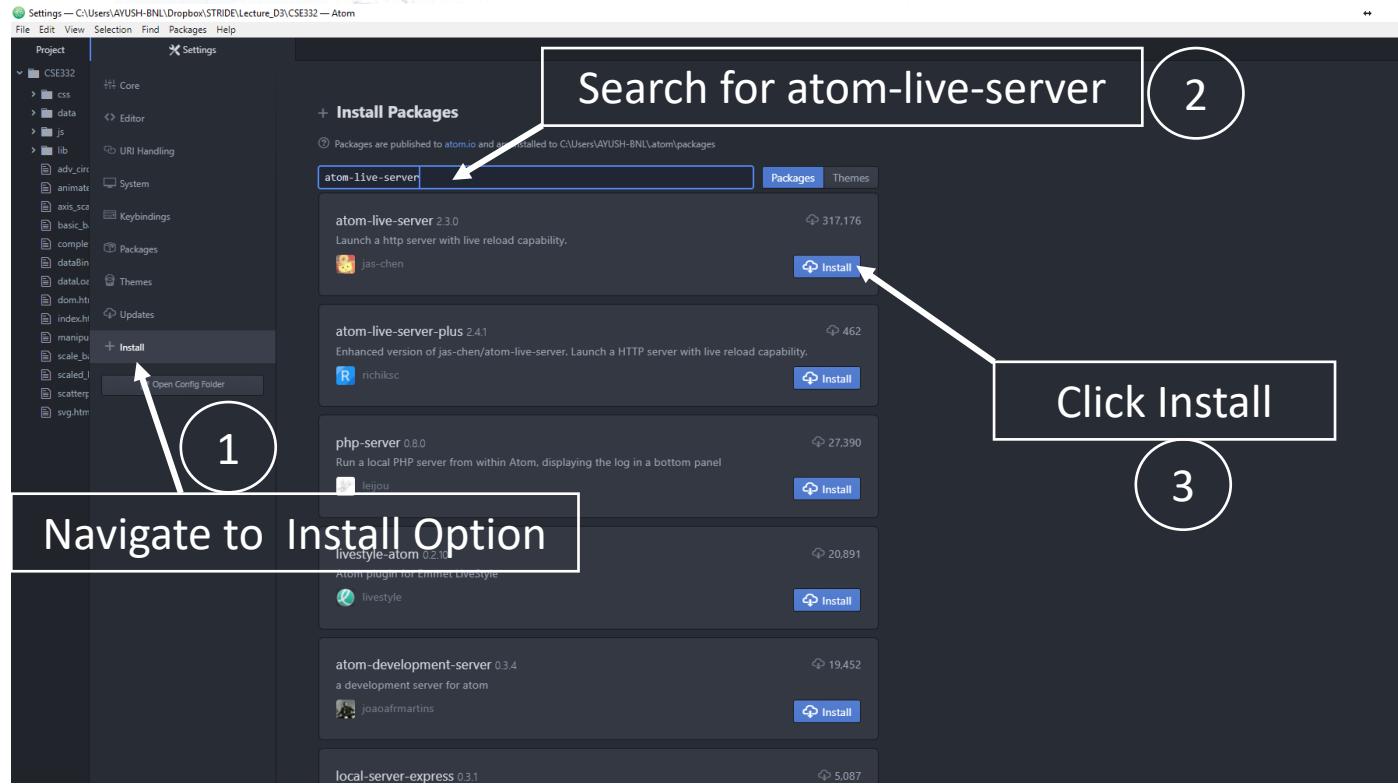
- D3 library
- Web server
- Editor
- Web browser



Editor Set Up



Editor Set Up



Done with Installation, Now Use below Keys in Editor to Launch the Server

ctrl-alt-1 launch live server on port 3000.

4

ctrl-alt-q stop live server.

ctrl-alt-3 launch live server on port 3000.

ctrl-alt-4 launch live server on port 4000.

ctrl-alt-5 launch live server on port 5000.

ctrl-alt-8 launch live server on port 8000.

ctrl-alt-9 launch live server on port 9000.

D3 Begins...

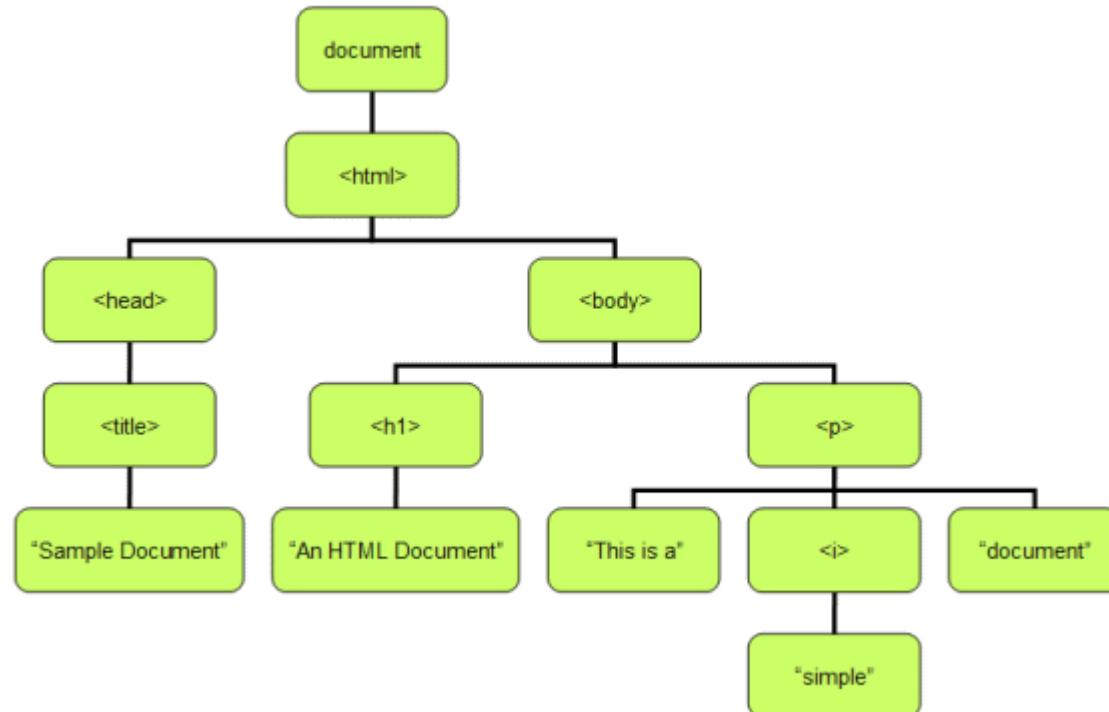
Setup D3.js Development Environment

- D3 library
- Web server
- Editor
- Web browser



D3 DOM

- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION



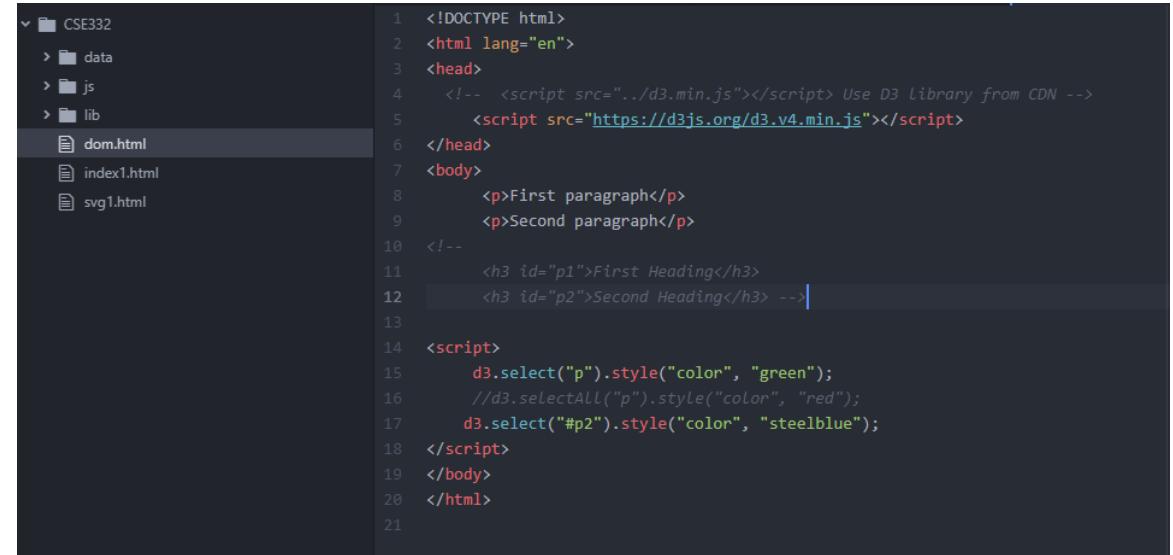
DOM Selection

- **d3.select()**

- Select Element By Tag Name
- Select Element By Id

- **d3.selectAll()**

- Select Element By Tag Name
- Select Element By Id
- Select Element By CSS Class Name



The image shows a code editor interface with a dark theme. On the left, there is a file tree with the following structure:

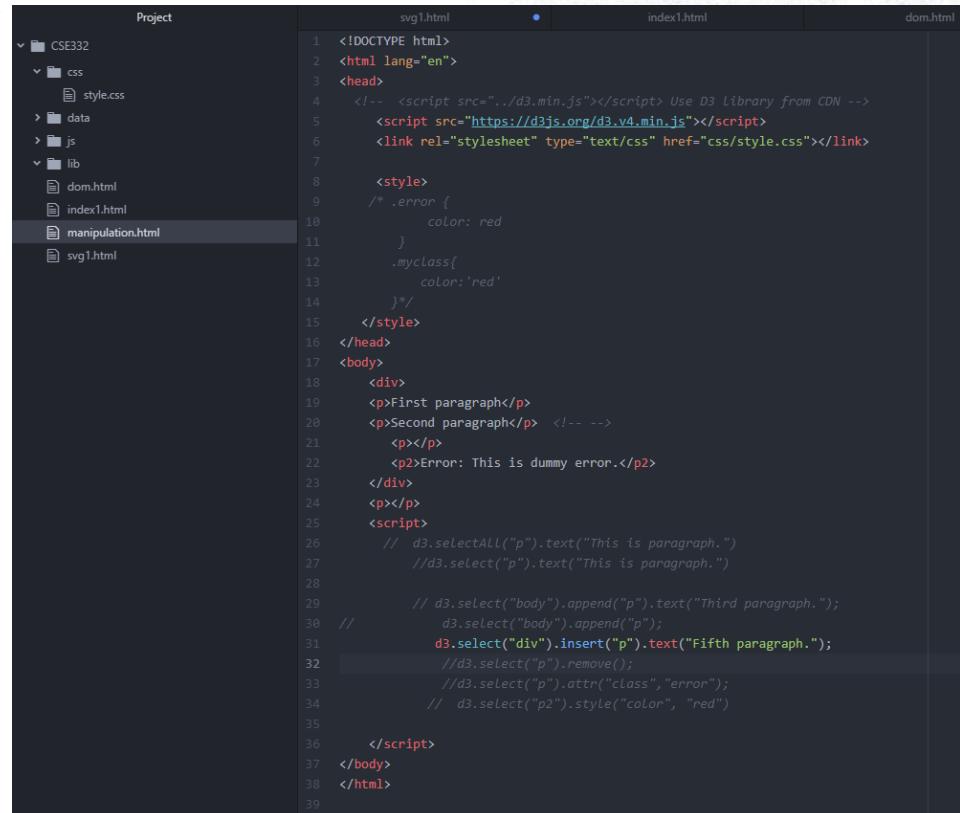
- CSE332
 - data
 - js
 - lib
 - dom.html** (highlighted)
 - index1.html
 - svg1.html

The main pane displays a script block containing D3.js code. The code uses the `d3.select` and `d3.selectAll` methods to style elements in the DOM.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <!-- <script src="../d3.min.js"></script> Use D3 Library from CDN -->
5      <script src="https://d3js.org/d3.v4.min.js"></script>
6  </head>
7  <body>
8      <p>First paragraph</p>
9      <p>Second paragraph</p>
10     <!--
11         <h3 id="p1">First Heading</h3>
12         <h3 id="p2">Second Heading</h3> --|
13     -->
14     <script>
15         d3.select("p").style("color", "green");
16         //d3.selectAll("p").style("color", "red");
17         d3.select("#p2").style("color", "steelblue");
18     </script>
19  </body>
20 </html>
```

DOM Manipulation

- `text()`
- `append()`
- `insert()`
- `remove()`
- `html()`
- `attr()`
- `property()`
- `style()`



The image shows a code editor interface with a dark theme. On the left, there is a 'Project' sidebar listing files and folders: CSE332, css (style.css), data, js, lib (dom.html, index1.html, manipulation.html), and svg1.html. The main area displays the content of manipulation.html. The code is a combination of HTML and D3.js script tags.

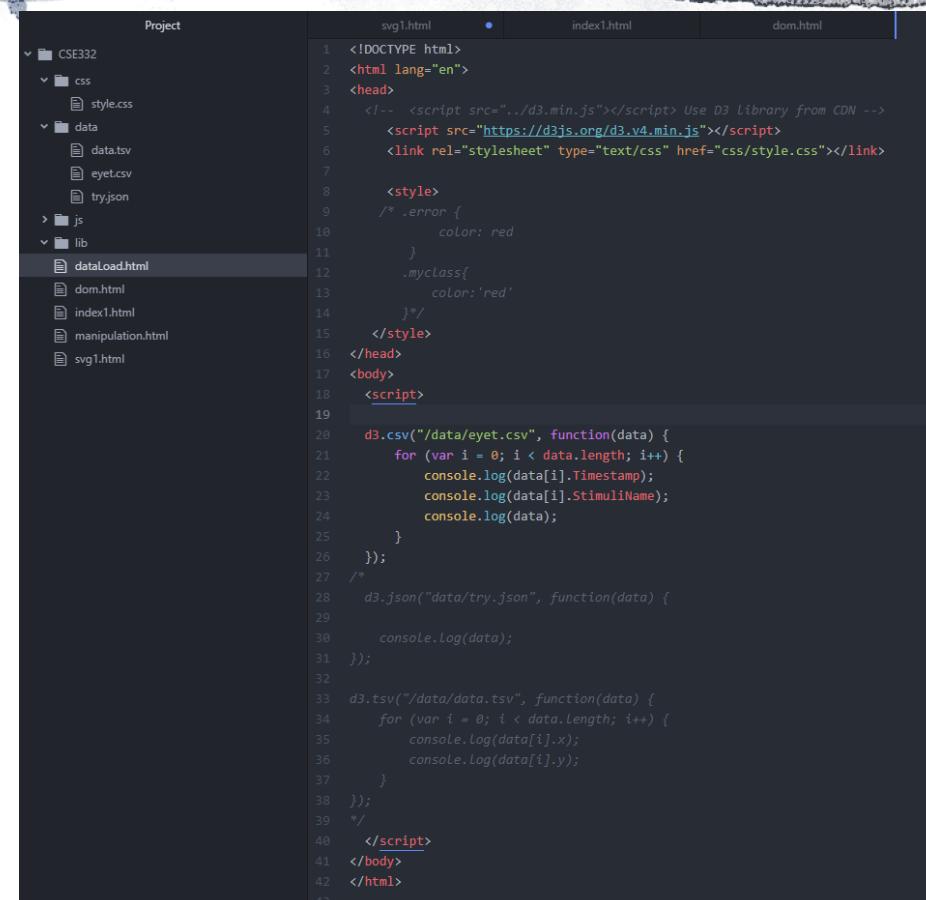
```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- <script src="../d3.min.js"></script> Use D3 Library from CDN -->
<script src="https://d3js.org/d3.v4.min.js"></script>
<link rel="stylesheet" type="text/css" href="css/style.css"></link>
<style>
/* .error {
color: red
}
.myclass{
color: 'red'
} */
</style>
</head>
<body>
<div>
<p>First paragraph</p>
<p>Second paragraph</p> <!-- -->
<p></p>
<p>Error: This is dummy error.</p>
</div>
<p></p>
<script>
// d3.selectAll("p").text("This is paragraph.")
//d3.select("p").text("This is paragraph.")

// d3.select("body").append("p").text("Third paragraph.");
//d3.select("body").append("p");
d3.select("div").insert("p").text("Fifth paragraph.");
//d3.select("p").remove();
//d3.select("p").attr("class", "error");
// d3.select("p2").style("color", "red")

</script>
</body>
</html>
```

D3 Data Loading

- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION



The image shows a code editor interface with a dark theme. On the left is a 'Project' sidebar listing files and folders: CSE332, css (style.css), data (data.tsv, eyet.csv, try.json), js, lib, and several HTML files (dataLoad.html, dom.html, index1.html, manipulation.html, svg1.html). The main pane displays a script file named 'svg1.html'. The code in the script file is:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <!-- <script src="../d3.min.js"></script> Use D3 library from CDN -->
5      <script src="https://d3js.org/d3.v4.min.js"></script>
6      <link rel="stylesheet" type="text/css" href="css/style.css"></link>
7
8      <style>
9          /* .error {
10              color: red
11          }
12          .myClass{
13              color: 'red'
14          }*/
15      </style>
16  </head>
17  <body>
18      <script>
19
20          d3.csv("/data/eyet.csv", function(data) {
21              for (var i = 0; i < data.length; i++) {
22                  console.log(data[i].Timestamp);
23                  console.log(data[i].StimuliName);
24                  console.log(data);
25              }
26          });
27      /*
28          d3.json("data/try.json", function(data) {
29
30              console.log(data);
31          });
32
33          d3.tsv("/data/data.tsv", function(data) {
34              for (var i = 0; i < data.length; i++) {
35                  console.log(data[i].x);
36                  console.log(data[i].y);
37              }
38          });
39      */
40      </script>
41  </body>
42  </html>
43
```

D3 Data Loading

- d3.csv()

| A | B | C | D | E | F | G | H | I | J |
|----|-----------|-------------|---------------|------------------|----------------|---------------|-------------|---|---|
| 1 | Timestamp | StimuliName | FixationIndex | FixationDuration | MappedFixation | MappedFixUser | description | | |
| 2 | 2586 | 01_Antwe | 9 | 250 | 1151 | 458 p1 | color | | |
| 3 | 2836 | 01_Antwe | 10 | 150 | 1371 | 316 p1 | color | | |
| 4 | 2986 | 01_Antwe | 11 | 283 | 1342 | 287 p1 | color | | |
| 5 | 3269 | 01_Antwe | 12 | 433 | 762 | 303 p1 | color | | |
| 6 | 3702 | 01_Antwe | 13 | 183 | 624 | 297 p1 | color | | |
| 7 | 3885 | 01_Antwe | 14 | 333 | 712 | 303 p1 | color | | |
| 8 | 4218 | 01_Antwe | 15 | 300 | 753 | 293 p1 | color | | |
| 9 | 4518 | 01_Antwe | 16 | 516 | 804 | 284 p1 | color | | |
| 10 | 5035 | 01_Antwe | 17 | 183 | 724 | 305 p1 | color | | |
| 11 | 5218 | 01_Antwe | 18 | 250 | 652 | 703 p1 | color | | |
| 12 | 5468 | 01_Antwe | 19 | 183 | 495 | 855 p1 | color | | |
| 13 | 5651 | 01_Antwe | 20 | 550 | 425 | 976 p1 | color | | |

data.tsv

| | x | y |
|---|----|----|
| 1 | 5 | 90 |
| 2 | 25 | 30 |
| 3 | 45 | 50 |
| 4 | 65 | 55 |
| 5 | 85 | 25 |

```
<?xml version = "1.0" encoding = "utf-8"?>
<!-- xsplane.1.xml -->
<xslstylesheet type = "text/xsl" href = "xsplane.1.xsl" ?>
<plane>
  <year> 1977 </year>
  <make> Cessna </make>
  <model> Skyhawk </model>
  <color> Light blue and white </color>
</plane>
```

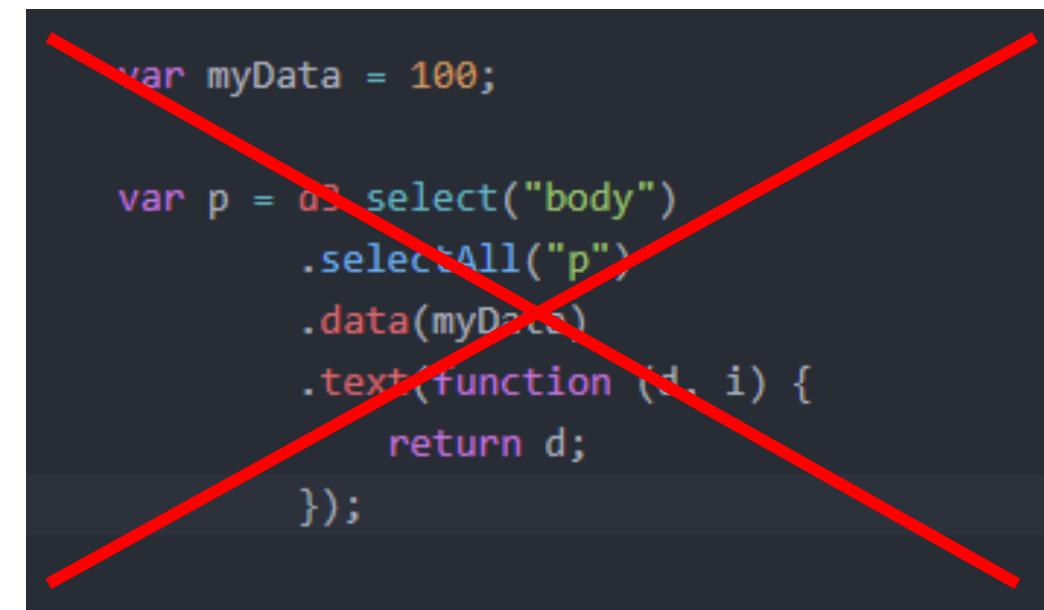
```
    },
    {
      color: "magenta",
      value: "#f0f"
    },
    {
      color: "yellow",
      value: "#ff0"
    },
    {
      color: "black",
      value: "#000"
    }
]
```

D3 DOM

- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION

D3 Data Binding

- data()
- enter()
- exit()
- datum()



```
var myData = 100;

var p = d3.select("body")
    .selectAll("p")
    .data(myData)
    .text(function (d, i) {
        return d;
});
```

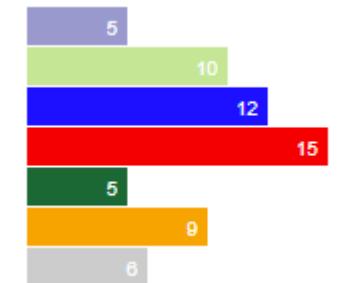
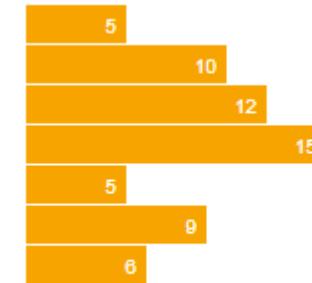
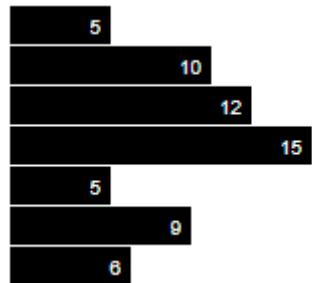
D3 Data Binding

- data()
- enter()
- exit()
- datum()

```
var data = [4, 1, 6, 2, 8, 9];
var body = d3.select("body")
    .selectAll("span")
    .data(data)
    .enter()
    .append("span")
    .text(function(d) { return d + " "; });
```

D3 CHARTS

DOM SELECTION & MANIPULATION



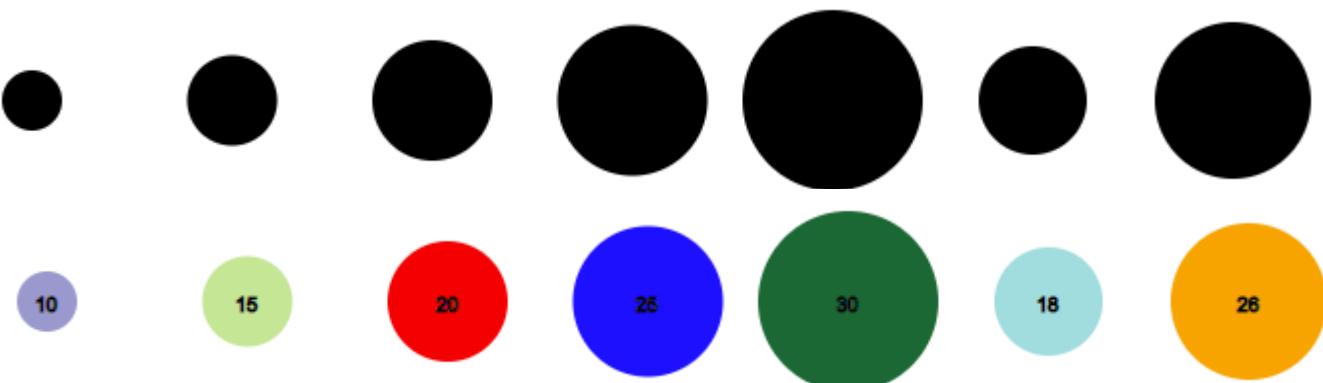
ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)

CHARTS

SCALES

AXES

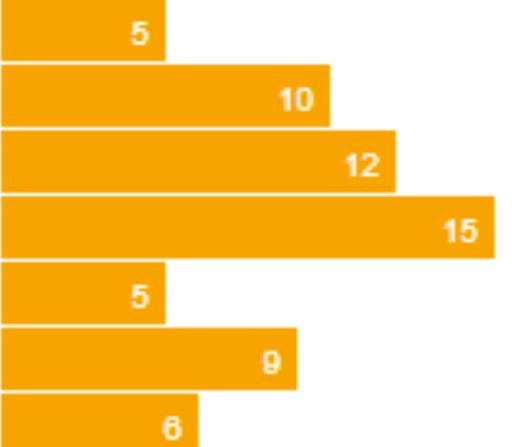
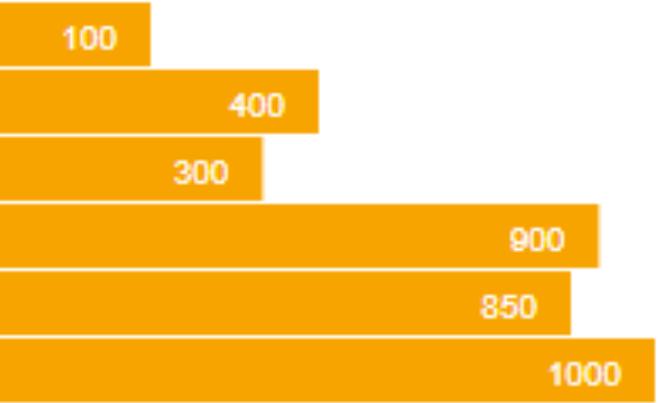
TRANSITIONS AND INTERACTION



D3 SCALING

- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION

D3 SCALING

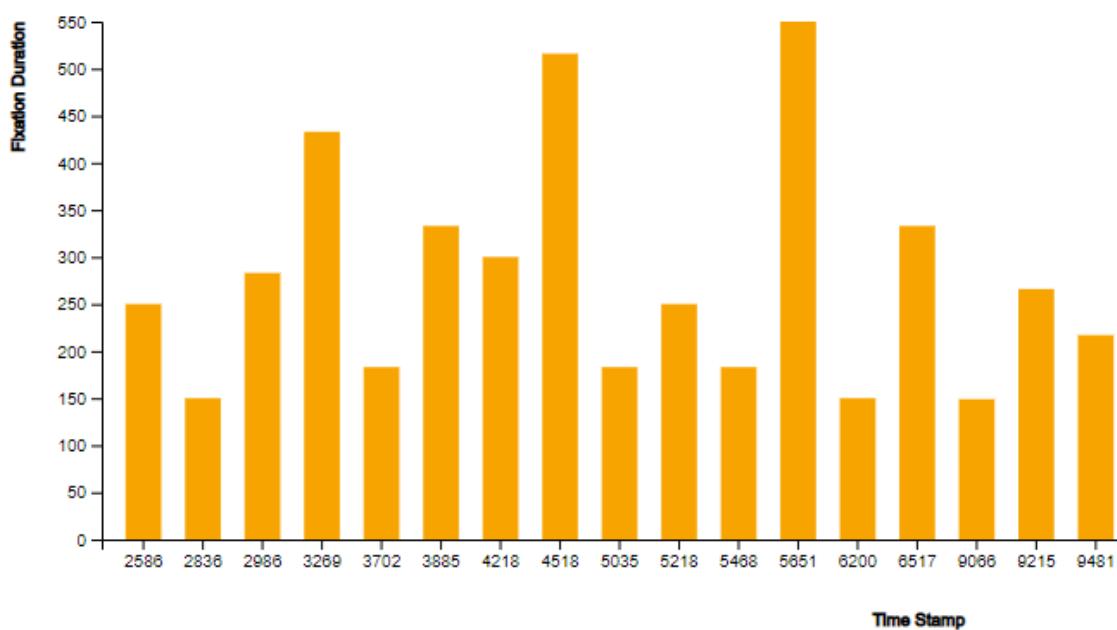
| Scale Type | Method | Description |
|------------|--|--|
| Continuous | d3.scaleLinear() | Construct continuous linear scale where input data (domain) maps to specified output range. |
| | d3.scaleIdentity() | Construct linear scale where input data is the same as output. |
| Sequential |  5 10 12 15 | where i scale. scale. scale. cale wh ale with le wher arbitra |
| Quantile |  5 9 | scale. scale. cale wh ale with le wher arbitra |
| Quantile |  6 8 | scale. scale. cale wh ale with le wher arbitra |
| Threshold | | arbitra |
| Band | | ordinal scales except the output range is continuous and numeric. |
| Point | d3.scalePoint() | Construct point scale. |
| Ordinal | d3.scaleOrdinal() | Construct ordinal scale where input data includes alphabets and are mapped to discrete numeric output range. |

D3 Axis

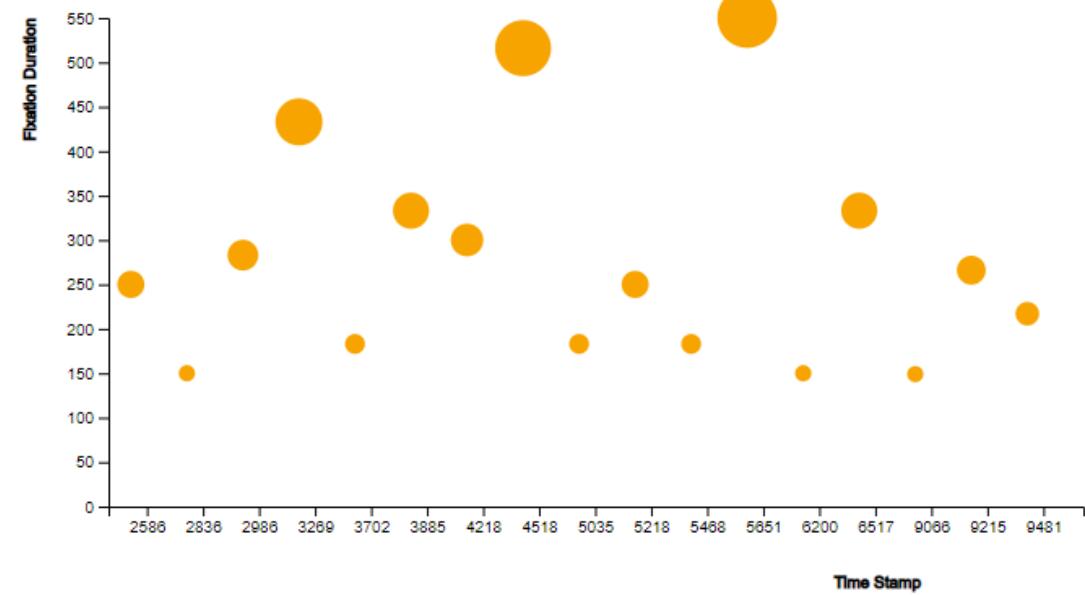
- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION

D3 Axis (Exactly Same Data)

Eye Movement Data



Eye Movement Data

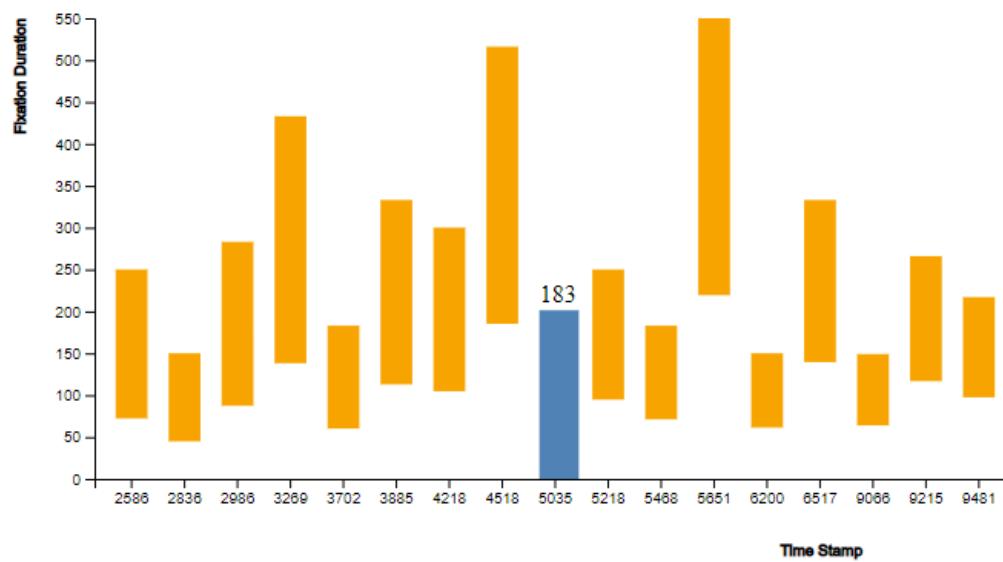


D3 Animation

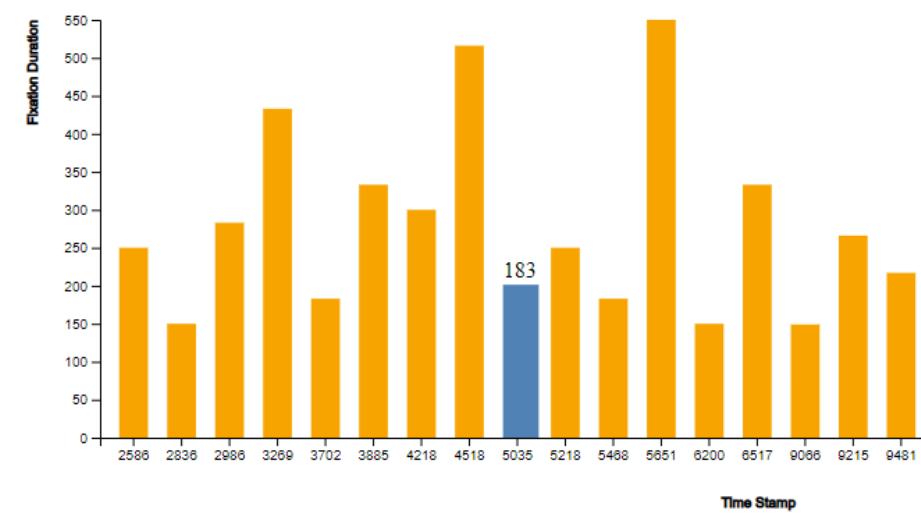
- DOM SELECTION & MANIPULATION
- LOADING DATA
- ENTER-UPDATE-EXIT PARADIGM (DATA BINDING)
- CHARTS
- SCALES
- AXES
- TRANSITIONS AND INTERACTION

D3 Animation

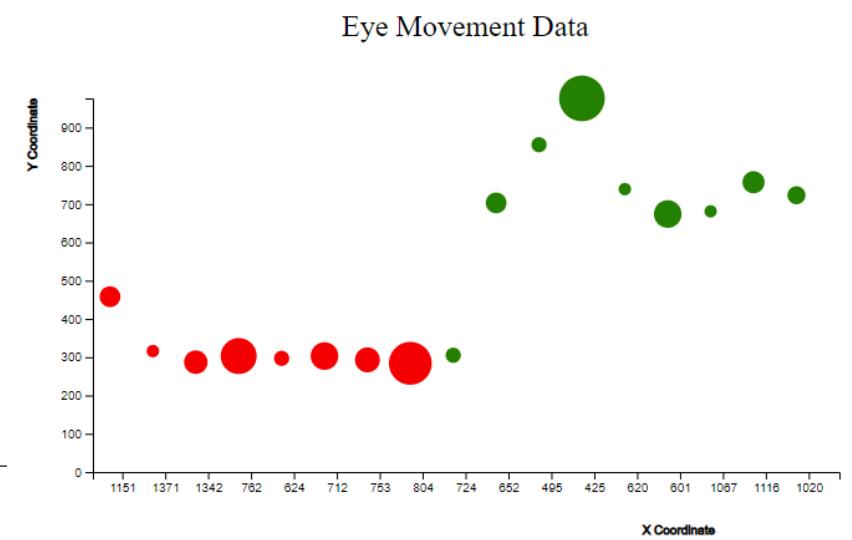
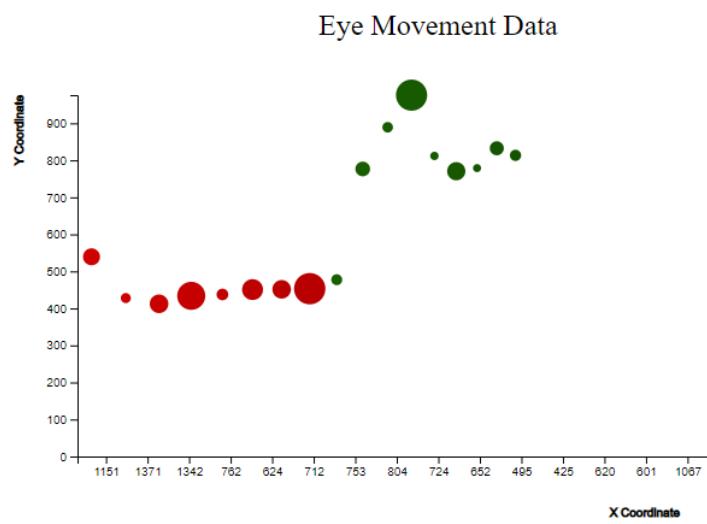
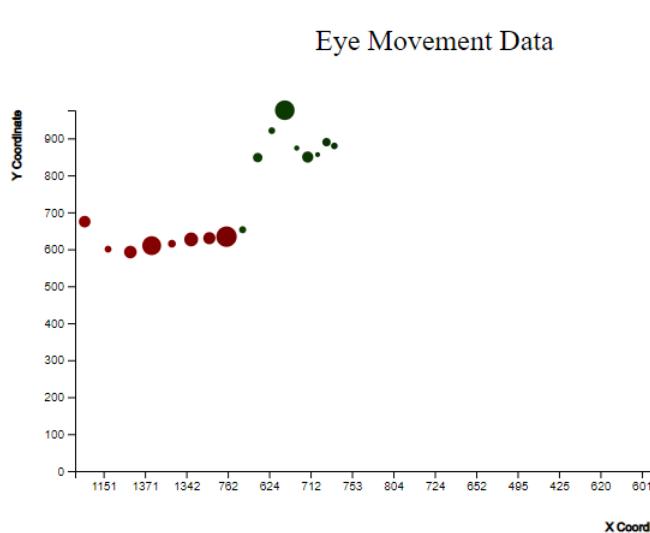
Eye Movement Data



Eye Movement Data



D3 Animation



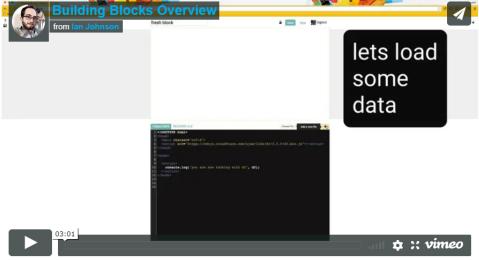
Block Builder

[START CODING](#)

Block Builder

QUICKLY CREATE, EDIT AND FORK D3JS EXAMPLES

Are you learning d3 or trying out new ideas? Block Builder is an in-browser code editor built for creating and sharing d3.js examples. Check out this short video for an overview of how it works!



Create and Edit

If you login with GitHub, all of your examples will save to GitHub gists associated with your account. Everything is powered by URL, so when you create a new block in Block Builder your

5:24 PM
2/5/2019

[fresh block](#)

[Save](#) [New](#) [login](#)

`index.html README.md ... (1)`

Edit the code below to change me!

```
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<script src="https://d3js.org/d3.v4.min.js"></script>
<style>
body { margin:0;position:fixed;top:0;right:0;bottom:0;left:0; }
</style>
</head>
<body>
<script>
// Feel free to change or delete any of the code you see in this editor!
</script>
```

<https://blockbuilder.org/>

References

- <https://www.dashingd3js.com/d3-v4-tutorial>
- <https://github.com/d3/d3/wiki/Gallery>
- <http://poloclub.gatech.edu/cse6242/2014spring/lectures/CSE6242-20140123-Stolper-D3.pdf>
- <https://bost.ocks.org/mike/d3/workshop/#0>
- <http://www.tutorialsteacher.com/d3js>
- <https://square.github.io/intro-to-d3/>
- <http://alignedleft.com/tutorials>
- <https://www.tutorialspoint.com/d3js/>