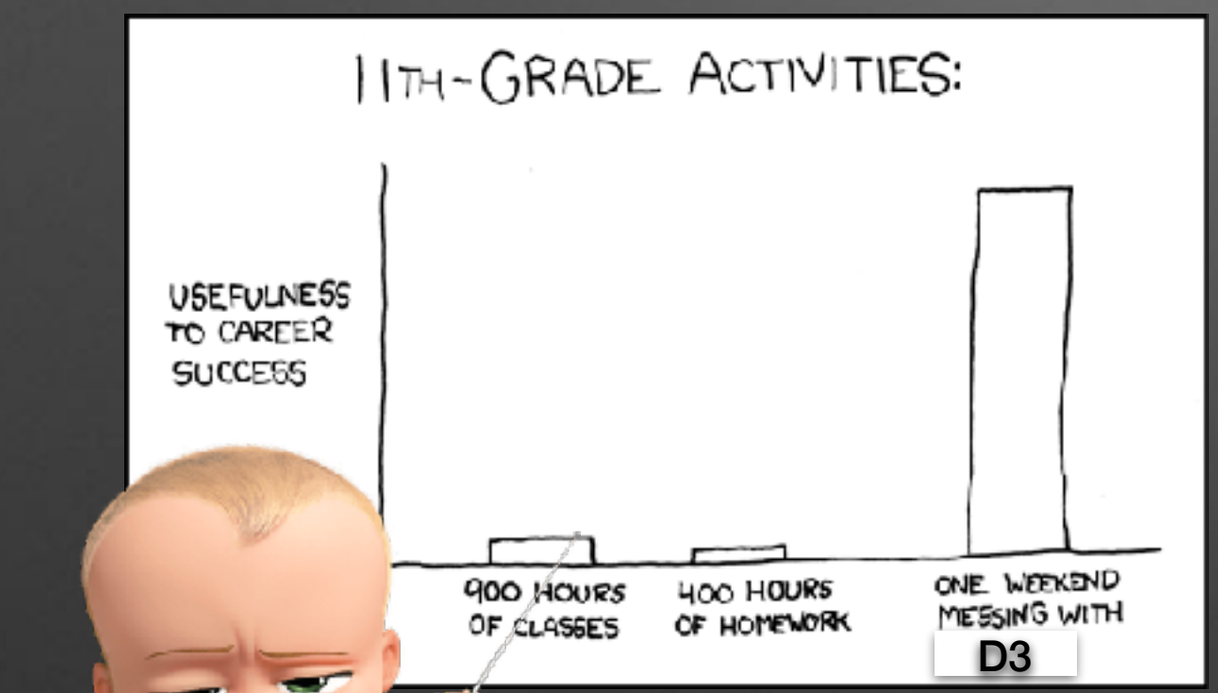
# Intro to D3.js

By Andrew Chumich

@AndrewChumich

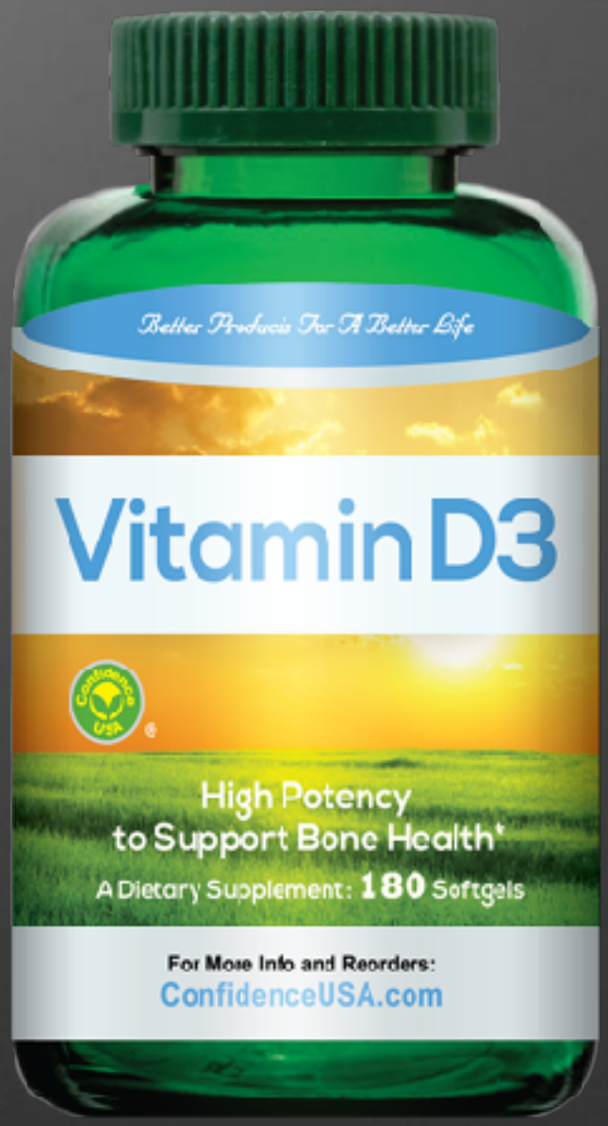
# What is it?




- Data-Driven Documents (D3)
- Data-driven DOM manipulation & transformation
  - jQuery + data binding + much more
- Charts!
- Maps!
- Any data visualization you can dream up

# Important Concepts

- Selection
  - Grab your DOM elements
- Data binding
  - Bind the elements to data
- Exit, enter, update pattern
  - Repeatable render function



# Selection

```
<div class="viz">
    <svg>
        <g class="bar-chart">
            <g class="bar-group">
                <rect x="10" y="10" width="100" height="100"/>
            </g>
            <g class="bar-group">
                <rect x="110" y="10" width="100" height="70"/>
            </g>
            <g class="bar-group">
                <rect x="210" y="10" width="100" height="80"/>
            </g>
        </g>
    </svg>
</div>
```

```
1   let barChartGroup = d3.select('g.bar-chart');
2
3   let selection = barChartGroup
4       .selectAll('g.bar-group');
5
6
7
8
```

- Select already existing elements
- But what if I want to add new elements/data?

# Data Binding

- Use `.data([data [, key]])` to bind data to a selection
- This works even though no `g.bar-group` elements exist yet
- Returns `update` selection

```
<div class="viz">
    <svg>
        <g class="bar-chart">
        </g>
    </svg>
</div>
```

```
let barChartGroup = d3.select('g.bar-chart');

const DATA_ARRAY = [
    {
        id: 0,
        value: 1,
    },
    {
        id: 1,
        value: 2,
    },
    {
        id: 2,
        value: 8,
    },
    {
        id: 3,
        value: 4,
    },
];

let update = barChartGroup
    .selectAll('g.bar-group')
    .data(DATA_ARRAY, function(d) {
        // if you don't do this, it will key by index
        return d.id;
    });
```

# Exit, Enter, Update

- You can get the `exit` and `enter` selections from the `update` selection

- Doing all 3 of these in the same function makes for a handy, repeatable render function

```
let update = barChartGroup
    .selectAll('g.bar-group')
    .data(DATA_ARRAY, function(d) {
        // if you don't do this, it will key by index
        return d.id;
    });

update
    // get the exit selection
    .exit()
    // remove exiting elements
    .remove()
    ;


update
    // get the enter selection
    .enter()
    // append g for each new item and add attributes
    .append('g')
        .attr('class', 'bar-group')
        .append('rect')
        // set height, width, color, position
        // .attr('width', xScale.bandwidth())
        // ...
    ;

update
    // update height, width, color, position
    // .attr('width', xScale.bandwidth())
    // ...
```

# Now for some real code!

https://andrewchumich.github.io/d3-zerrtech/
https://github.com/andrewchumich/d3-zerrtech

# D3 Resources

- d3js.org official docs
- https://bl.ocks.org/mbostock/3808218 intro to enter, exit, update pattern
- https://bl.ocks.org/mbostock a bunch of live examples