

Questions D3 Intro:

How can D3 access and change the DOM? What do `select` and `selectAll` do?

D3 accesses the dom by `d3.select()` or `d3.selectAll()` with an element inside. After this you can do stuff with that element.

Select takes the first instance it finds of the particular element you want, while select all takes all elements specified.

What are the `d` and `i` in `function(d){}` and `function(d, i){}`?

The `d` is just a convention, `d` is standard fare in d3 code because that's the writing style of Mike Bostock, the author of the framework and of many of its examples.

`i` is just a second argument, so again could be named anything. It behaves as a for-loop so index 0 is the first of the data in the data array, 1 is the second, etc.

Write sample lines of JavaScript to add a `div` element with class `"barChart1"` and to add an `svg` element with class `"barChart2"` with square dimensions.

```
d3.select("body").append("div").attr("class", "BarChart1");
```

```
d3.select("body").append("svg")
    .attr("class", "BarChart2")
    .attr("width", 250)
    .attr("height", 250);
```

Describe `append`, `update`, `enter`, and `exit` at a high level. What does `"selectAll + data + enter + append"` refer to?

Its binding data, it selects all elements specified between the `selectAll()` brackets, then for each data point you enter the element where there is no data element in the dom, and then append it. Like the following:

```
Var data = (1,2,3,4,5);
```

```
Var scale = d3.scale.linear().domain([1,5]).range([0,200]);
```

```
var svg = d3.select("body").append("svg").attr("class", "BarChart2").attr("width",
250).attr("height", 250);
```

```
svg.selectAll("rect")
    .data(data)
    .enter().append("rect")
```

```
.attr("x", scale)
.attr("y", 50)
.attr("width", 20)
.attr("height", 20);
```

This code selects all the rects, then enters each data point in data and enters each rect that doesn't already exist and then appends a rect there that is 20 by 20 pixels, at a y of 50 and an x that will change each new rect with a fixed distance between 0 and 200 with steps of $1/5^{\text{th}}$ of that range.

Or as they say at the bar chart part 1 site:

The data operator returns the update selection. The enter and exit selections hang off the update selection, so you can ignore them if you don't need to add or remove elements.

What are the main differences between drawing a bar chart with HTML and SVG?

In HTML you would have to hardcode your chart so whenever you change your dataset you have to redo the whole bar chart, while with SVG you can dynamically create the bar chart. Furthermore it generally takes a lot more code to make a HTML bar chart than with SVG. Plus as the name scalable vector graphic might suggest, an SVG can be scaled without much problems, while the HTML bar chart only has the amount of pixels to work with that is has been given in the start.

In drawing the simple bar chart with D3 and SVG, what elements were appended, and what parts of the graph did these elements correspond to?

First it selects a div with class="chart", then appends another div with a desired width. This would be the outline of the chart. Then for each bar.data (data) (or barUpdate) another div is appended, so that would be all the bars inside the bar chart.