Intro to D3 and Representing Datacenter with Tree-view in D3

By:
Sushil Verma
Team: VRB / Symphony

August 31, 2017

About D3 (Data-Driven Documents)

❑ Open-source JavaScript library developed by Mike Bostock.

❑ Used in Web Development and Data Analytics for Interactive Data Visualizations.

❑ Uses Web Standards: HTML, CSS, DOM, JavaScript, SVG.

❑ Data Driven: Array, Object, CSV, JSON, XML

❑ Custom Visualizations and Transitions: transition()

❑ Interaction and Animations: duration(), delay(), ease()

SVG (Scalable Vector Graphics)

- Not a direct image but is just a way to create images using text.

- Scales itself according to the size of the browser.

- Like styling html elements, styling SVG elements is simple.

- Most used SVG elements: Line, Rectangle, Circle, Ellipse, Text, Path, g(Group multiple nodes)

```
<svg width="500" height="500">
    <rect x="0" y="0" width="300" height="200" fill="yellow"></rect>
</svg>
```

DOM Selection using D3

- Include D3:

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

- D3 object and d3.select() method:

```
d3.select("p").style("color", "green");
d3.select("#p2").style("color", "green");
d3.select(".myclass ").style('color','green');
```

- d3.selectAll() method and nesting:

```
d3.selectAll(".myclass ").style('color','green');
d3.select("tr").selectAll("td").style('background-color','yellow');
```

DOM Manipulation using D3

- text("content"), append("element name"), insert("element name"), remove(), html("content")

- attr("name", "value"), property("name", "value"), style("name", "value"), classed("class", bool)

```
d3.select("p").text("This is paragraph.")
d3.select("body").append("p");
d3.select("body").append("p").text("Third paragraph.");
d3.select("div").insert("p").text("Second paragraph.");
d3.select("p").remove();
d3.select("p").html("<span>This was added in HTML</span>");
d3.select("p").attr("class","error");
d3.select("input").property("checked",true);
d3.select("p").style("color", "red")
d3.select("p").classed('error', true);
```

Functions of Data

❑ Can access 3 parameters inside function: d (data), i (index), this (current DOM)

```
var data = [1, 2, 3];
var paragraph = d3.select("body")
    .selectAll("p")
    .data(data)
    .text(function (d, i) {
            console.log("d: " + d);
            console.log("i: " + i);
            console.log("this: " + this);
            return d;
    });
```

Event Handling

- ❑ Can bind an event listener to any DOM element using d3.selection.on() method

```
d3.selectAll("div")
    .on("mouseover", function(){
          d3.select(this)
          .style("background-color", "orange");
          // Get current event info
          console.log(d3.event);
          // Get x & y co-ordinates
          console.log(d3.mouse(this));
    })
    .on("mouseout", function(){
          d3.select(this)
          .style("background-color", "steelblue")
    });
```

Animation in D3

- ❑ Simplifies the process of animations with transitions.

- ❑ selection.transition(), transition.duration(), transition.ease(), transition.delay()

```
function update() {
    bar1.transition()
          .ease(d3.easeLinear)
          .duration(2000)
          .attr("height",100);

    bar2.transition()
          .ease(d3.easeLinear)
          .duration(2000)
          .delay(2000)
          .attr("height",100);
}
```

Data Loading and Binding

- Data Loading: d3.csv(), d3.json(), d3.tsv(), d3.xml()

```
d3.csv("/data/employees.csv",
          function(data) {});
```

- Data Binding: data(), enter(), exit(), datum()

```
var data = [10, 11, 20, 21, 30, 31];
var body = d3.select("body")
          .selectAll("span")
          .data(data)
          .enter()
          .append("span")
          .style('color', function(d) {
                if (d % 2 === 0) return "green";
                else return "red";
          })
          .text(function(d) { return d + " "; });
```

Bar Chart in D3

```
var data = [5, 10, 12, 18, 20];
var width = 200, scaleF = 10, barHeight = 20;

var graph = d3.select("body").append("svg").attr("width", width)
              .attr("height", barHeight * data.length);
var bar = graph.selectAll("g").data(data).enter().append("g")
               .attr("transform", function(d, i) {
                   return "translate(0," + i * barHeight + ")";
                });

bar.append("rect").attr("width", function(d) { return d*scaleF; })
   .attr("height", barHeight - 1);
bar.append("text").attr("x", function(d) { return (d*scaleF); })
   .attr("y", barHeight / 2)
   .attr("dy", ".35em").text(function(d) { return d; });
```

Scales and Axes in D3

```
var width = 400, height = 100;
var data = [10, 15, 20, 25, 30];
var svg = d3.select("body").append("svg").attr("width", width)
            .attr("height", height);

var xscale = d3.scaleLinear().domain([0, d3.max(data)])
               .range([0, width - 100]);
var yscale = d3.scaleLinear().domain([0, d3.max(data)])
               .range([height/2, 0]);

var x_axis = d3.axisBottom().scale(xscale);
var y_axis = d3.axisLeft().scale(yscale);
svg.append("g").attr("transform", "translate(50, 10)").call(y_axis);

var xAxisTranslate = height/2 + 10;
svg.append("g").attr("transform", "translate(50, " +
                 xAxisTranslate  +")").call(x_axis);
```

Pie and Donut Chart in D3

```
var data = [2, 4, 8, 10];
var svg = d3.select("svg"), width = svg.attr("width"),
   height = svg.attr("height"), radius = Math.min(width, height)/2,
   g = svg.append("g").attr("transform", "translate(" + width/2 +
       "," + height/2 + ")");
var color = d3.scaleOrdinal(['#4daf4a','#377eb8',
          '#ff7f00','#984ea3','#e41a1c']);

var pie = d3.pie();
var arc = d3.arc().outerRadius(radius);
                  .innerRadius(0);  // make it positive for Donut Chart
var arcs = g.selectAll("arc").data(pie(data)).enter()
            .append("g").attr("class", "arc")

arcs.append("path")
    .attr("fill", function(d, i) { return color(i);})
    .attr("d", arc);
```

Thank you