

# D3.js Quick Introduction

Paulo Dias



departamento de electrónica telecomunicações e informática



universidade de aveiro

## Introducing d3



- JavaScript library for creating data visualizations
- Data-Driven Documents
  - User provides the **data**
  - D3 does the **driving**
  - I.e., it **connects** the data to **web-based documents**
- Mike Bostock
- [d3js.org](http://d3js.org)

2

## Introducing d3



- Generation and manipulation of web-documents with data
- How ?
  - **Load** the data into the browser's **memory**
  - **Bind** the data to document **elements**
  - **Transform** elements (i.e., set **visual properties**)  
according to each element's bound datum
  - **Transition** elements between states in response to **user**  
**input**

3

## Introducing D3



- The transformation step is the most important !
  - **Mapping rules**
    - Should larger values make taller bars? Or brighter circles?
    - What color palette is used to fill regions in a map?
    - ...
- **Visual decisions** are left to the visualization designer !
  - D3 executes them !

## Explanatory vs. Exploratory Visualization



- **Exploratory** Visualization
  - Discover **significant, meaningful patterns** in the data
  - Generate **multiple views** of the same data set
  - Tools: [Tableau](http://tableau.com), [ggplot2](http://ggplot2.org), ...
- **Explanatory** Visualization
  - Present a view of the data **that highlights what has already been discovered**
  - D3 excels at this !!

## Introducing d3



- No support for **older browsers**
- No handling of **bitmap** map tiles
  - **Vector graphics** instead !
- No hiding of **original data**
  - **Client-side** execution
  - Data must be sent to the client
  - Do not use D3 if your data cannot be shared !

6

## Introducing d3



- 1996: first browser with JavaScript
- 2005: J. Heer et al.'s [prefuse](#) toolkit
  - Bringing data visualization to the Web
  - Accessible to less-than-expert programmers
  - Java / Needs a Java plug-in
- 2007: J. Heer's [Flare](#) toolkit
  - ActionScript
  - Needs Adobe's Flash Player for viewing
- 2009: J. Heer + M. Bostock – [Protovis](#)
  - JavaScript-based visualization toolkit
  - Native browser technologies only !!
  - Abstract representation layer

7

## Introducing d3



- 2011: [D3](#)
  - Mike Bostock + Vadim Ogievetsky + Jeff Heer
  - It operates directly on **web-documents**
  - Easier **debugging** and **experimentation**
  - More **visual possibilities**
  - BUT, potentially **steeper learning curve**
- **HTML + CSS + JavaScript + SVG**

8

## D3 – Generating page elements



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3 Page Template</title>
    <script type="text/javascript" src="http://d3js.org/d3.v3.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      <!-- D3 Code here -->
    </script>
  </body>
</html>
```

Content Delivery Network (CDN)  
src="http://d3js.org/d3.v3.js"  
Locally:  
src="d3.min.js"

## SVG – Circle Drawing



```
var dataset = [5, 10, 15, 20, 25];
var w = 500;
var h = 50;
var svg = d3.select("body")
  .append("svg")
  .attr("width", w)
  .attr("height", h);

var circles = svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle");

circles.attr("cx", "10")
  .attr("cy", "10")
  .attr("r", "10");
```

Dataset

Create SVG element

Data Binding

Modify element attributes

## SVG – Circle Drawing



```
...
var circles = svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle");

circles.attr("cx", function(d, i) {
  return (i * 50) + 25;
})
  .attr("cy", h/2)
  .attr("r", function(d) {
    return d;
  });
```

function of data (called for each of our data values)

## SVG – Colorful circles



```
...
circles.attr("cx", function(d, i) {
  return (i * 50) + 25;
})
  .attr("cy", h/2)
  .attr("r", function(d) {
    return d;
  })
  .attr("fill", "yellow")
  .attr("stroke", "orange")
  .attr("stroke-width", function(d) {
    return d/2;
  });
```

## SVG – Creating a bar chart (1)



```
...  
  
var w = 500;  
var h = 100;  
var barPadding = 1;  
  
var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,  
               11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];  
  
var svg = d3.select("body")  
  .append("svg")  
  .attr("width", w)  
  .attr("height", h);
```

## SVG – Creating a bar chart (2)



```
svg.selectAll("rect")  
  .data(dataset)  
  .enter()  
  .append("rect")  
  .attr("x", function(d, i) {  
    return i * (w / dataset.length);  
  })  
  .attr("y", function(d) {  
    return h - (d * 4);  
  })  
  .attr("width", w / dataset.length - barPadding)  
  .attr("height", function(d) {  
    return d * 4;  
  })  
  .attr("fill", "teal");
```



## SVG – Creating a scatter plot

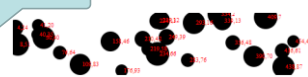


```
var w = 500;  
var h = 100;  
var dataset = [ [5, 20], [480, 90], [250, 50], [100, 33], [330, 95], [410, 12],  
               [475, 44], [25, 67], [85, 21], [220, 88] ];
```

```
var svg = d3.select("body")  
  .append("svg")  
  .attr("width", w)  
  .attr("height", h);  
  
svg.selectAll("circle")  
  .data(dataset)  
  .enter()  
  .append("circle")  
  .attr("cx", function(d) {  
    return d[0];  
  })  
  .attr("cy", function(d) {  
    return d[1];  
  })  
  .attr("r", 5);
```

SVG element is circle

Functions for attributes cx and cy



## D3 – What is next?



- **Scales**
  - Mapping an input domain to an output range
- **Axes**
- **Updates, Transitions and Motion**
  - Input data might change over time
  - Visual adjustments
- **Interactivity**
  - Listen to events
- **Layouts**
  - Generate data more convenient for a specific visual task

## SVG – Templates / Gallery



<https://github.com/d3/d3/wiki/Gallery>



## Reference



- Getting Started with D3, Mike Dewar, O'Reilly Media, June 2012
- Scott Murray, *Interactive Data Visualization for the Web*, O'Reilly Media, 2013
- <https://github.com/d3/d3/wiki/Tutorials>