

# D3.js

Seth House <seth@eseth.com>

Utah JavaScript

2013-04-16

# Outline

1 D3.js explained

2 D3.js basics

3 Resources



Last edited by gowalla, a day ago

# D3.js

- Mike Bostock
- Successor to Protoviz
- ~ September 2011

# Data-Driven Documents

- Bind arbitrary data to DOM elements
- Apply data-driven transformations to those DOM elements

# D3.js is not

- Not a custom drawing library
- Web standards
- HTML, SVG, CSS: required

# D3.js is

- A library
- Functional toolbox
- Mathematical toolbox
- Data structure toolbox

# D3.js does

- Declarative DOM manipulation
- Data binding
- Transitions, timers, shapes, ranges



# Outline

1 D3.js explained

2 **D3.js basics**

3 Resources

# D3.js basics

# Selections

```
<!doctype html>
<meta charset=utf-8>
<title>D3!</title>
<script src="http://d3js.org/d3.v3.min.js"></script>

<div></div>
```

Selections are arrays of elements:

```
var thing = d3.select('div');

var things = thing.selectAll('p');
```

## Binding data

Bind data to a selection:

```
var mythings = ['foo', 'bar', 'baz', 'qux'];
```

```
var thing = d3.select('div');
```

```
var things = thing.selectAll('p')  
    .data(mythings)  
    .enter()  
    .append('p')  
    ;
```

# Displaying data

```
var things = thing.selectAll('p')  
    .data(mythings)  
    .enter()  
    .append('p')  
    .text(function(d) { return d })  
    ;
```

# Using data

```
var things = thing.selectAll('p')
    .data(mythings)
    .enter()
    .append('p')
    .text(function(d) { return d })
    .classed('even', function(d, i) {
        return i % 2 === 0;
    })
    ;
```

# Data joins

```
// Select main element  
var thing = d3.select('div');  
// DATA JOIN (join new data with old elems)  
var things = main.selectAll('p').data(mythings);  
// UPDATE (old elems)  
.attr()  
  
// ENTER (create new elems)  
things.enter().append('p');  
// ENTER + UPDATE (entering and updating elems)  
.text(function(d) { return d })  
// EXIT (remove old elems)  
things.exit().remove();
```

## An update pattern

```
function update(sel) {  
  sel.each(function(data) {  
    var thing = d3.select(this);  
  
    var things = thing.selectAll('p')  
      .data(data)  
      .enter()  
      .append('p')  
      ;  
  });  
}  
  
d3.select('div')  
  .datum(mythings)  
  .call(update);
```



# Getting data

## Built-in ajax:

- Local
- `d3.xhr`, `d3.get`, `d3.post`
- `d3.json`
- `d3.csv`
- (more)

```
d3.json('/path/to/mythings.json',  
      function(error, data) {  
    d3.select('div')  
      .datum(data)  
      .call(update);  
  });
```

## Formatting data

```
var yields = [{yield: 48.87, variety: "Manchuria",
               year: 1931, site: "Waseca"}]
```

```
d3.nest()
  .key(function(d) { return d.year })
  .key(function(d) { return d.variety })
  .entries(yields);
```

```
[{ "key": "1931",
   "values": [{
     "key": "Manchuria",
     "values": [{"yield": 48.87,
                  "variety": "Manchuria",
                  "year": 1931, "site": "Waseca"}]}
  ]
}]
```

# Outline

1 D3.js explained

2 D3.js basics

3 **Resources**

# Resources

# Towards Reusable Charts

```
function chart() {  
  var default_stuffs;  
  
  function update(selection) {  
    selection.each(function(data) {  
    });  
  }  
  update.default_stuffs = function(val) {  
    if (!arguments.length) return default_stuffs;  
    default_stuffs = val;  
    return update;  
  };  
  
  return update;  
}
```

# Tributary

- `http://tributary.io/tributary`
- `http://tributary.io/inlet/<GIST_ID>/`

## D3 Freestyles

- <http://enjalot.github.io/dot-enter/>
- <http://enjalot.github.io/dot-append/>

# D3 Wiki

`https://github.com/mbostock/d3/wiki`

- Examples gallery
- Tutorials
- Plugins
- API reference



# Color

<http://www.jeromecukier.net/blog/2011/08/11/d3-scales-and-color/>

# Inspiration

`http://www.reddit.com/r/dataisbeautiful/`