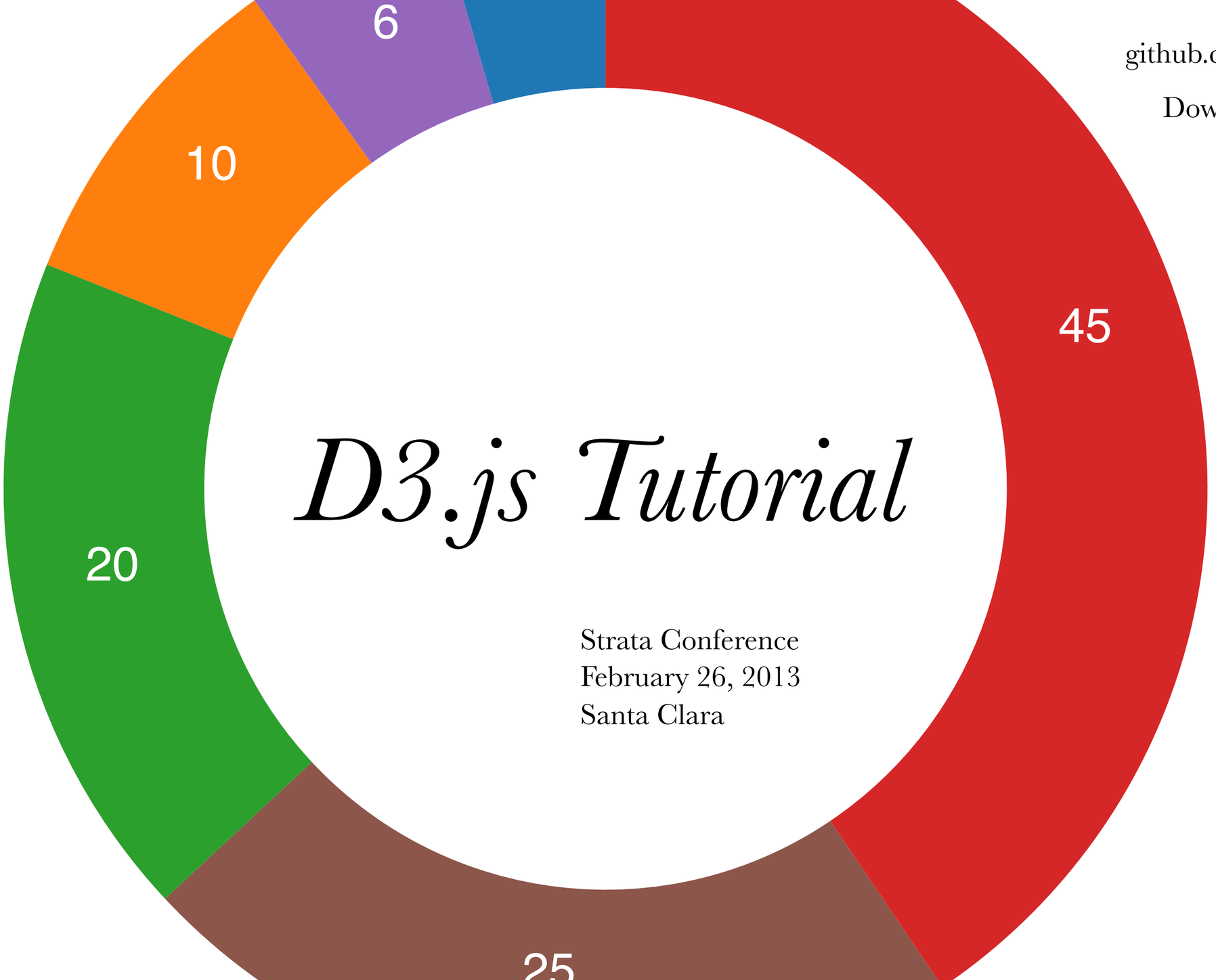
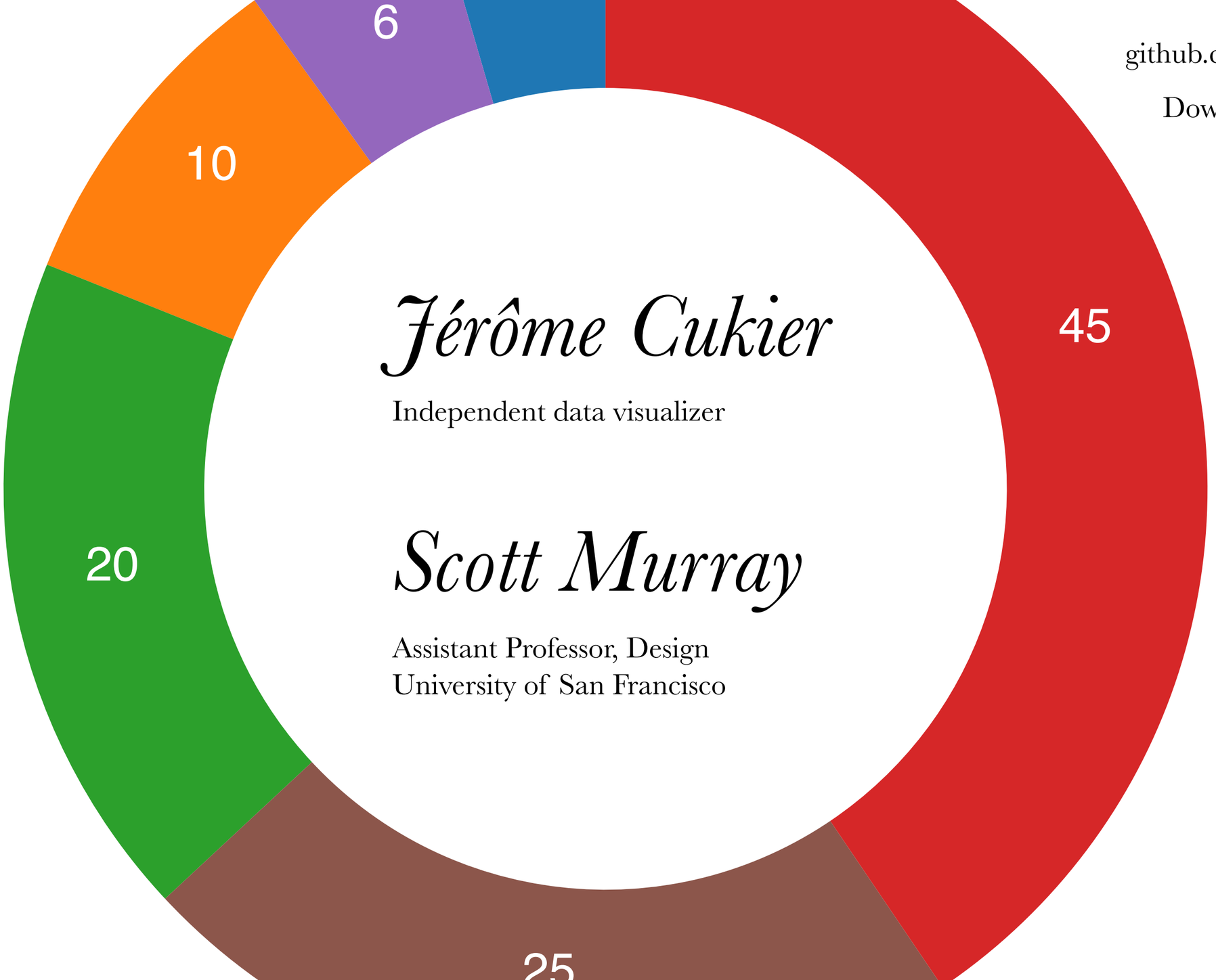


D3.js Tutorial

Strata Conference
February 26, 2013
Santa Clara

github.com/alignedleft/strata-d3-tutorial

Download the code examples!



Jérôme Cukier

Independent data visualizer

Scott Murray

Assistant Professor, Design
University of San Francisco

github.com/alignedleft/strata-d3-tutorial

Download the code examples!

What is d3.js?

```
d3 = function() {
  var π = Math.PI, ε = 1e-6, d3 = {
    version: "3.0.6"
  }, d3_radians = π / 180, d3_degrees = 180 / π, d3_document = document, d3_window = window;
  function d3_target(d) {
    return d.target;
  }
  function d3_source(d) {
    return d.source;
  }
  var d3_format_decimalPoint = ".", d3_format_thousandsSeparator = ",", d3_format_grouping = [ 3, 3 ];
  if (!Date.now) Date.now = function() {
    return +new Date();
  };
  try {
    d3_document.createElement("div").style.setProperty("opacity", 0, "");
  } catch (error) {
    var d3_style_prototype = d3_window.CSSStyleDeclaration.prototype, d3_style_setProperty = d3_style_prototype.setProperty;
    d3_style_prototype.setProperty = function(name, value, priority) {
      d3_style_setProperty.call(this, name, value + "", priority);
    };
  }
  function d3_class(ctor, properties) {
    try {
      for (var key in properties) {
        Object.defineProperty(ctor.prototype, key, {
          value: properties[key],
          enumerable: false
        });
      }
    } catch (e) {
      ctor.prototype = properties;
    }
  }
  var d3_array = d3_arraySlice;
  function d3_arrayCopy(pseudoarray) {
    var i = -1, n = pseudoarray.length, array = [];
    while (++i < n) array.push(pseudoarray[i]);
    return array;
  }
  function d3_arraySlice(pseudoarray) {
    return Array.prototype.slice.call(pseudoarray);
  }
  try {
    d3_array(d3_document.documentElement.childNodes)[0].nodeType;
  } catch (e) {
    d3_array = d3_arrayCopy;
  }
```

```
d3=function(){function t(t){return t.target}function n(t){return t.source}function e(t,n){try{for(var e in
n)Object.defineProperty(t.prototype,e,{value:n[e],enumerable:!1})}catch(r){t.prototype=n}}function r(t){for(var
n=-1,e=t.length,r=[];e>++n;)r.push(t[n]);return r}function u(t){return Array.prototype.slice.call(t)}function
i(){}function a(t){return t}function o(){return!0}function c(t){return"function"==typeof t?t:function(){return
t}}function l(t,n,e){return function(){var r=e.apply(n,arguments);return arguments.length?t:r}}function f(t)
{return null!=t&&!isNaN(t)}function s(t){return t.length}function h(t){return t.trim().replace(/\s+/g," ")}
function g(t){for(var n=1;t*n%1;)n*=10;return n}function p(t){return 1===t.length?function(n,e){t(null==n?
e:null)}:t}function d(t){return t.responseText}function m(t){return JSON.parse(t.responseText)}function v(t){var
n=Di.createRange();return n.selectNode(Di.body),n.createContextualFragment(t.responseText)}function y(t){return
t.responseXML}function M(){}function b(t){function n(){for(var n,r=e,u=-1,i=r.length;i>++u;)
(n=r[u].on)&&n.apply(this,arguments);return t}var e=[],r=new i;return n.on=function(n,u){var i,a=r.get(n);return
2>arguments.length?a&&a.on:(a&&(a.on=null,e=e.slice(0,i=e.indexOf(a)).concat(e.slice(i
+1)),r.remove(n)),u&&e.push(r.set(n,{on:u})),t)},n}function x(t,n){return n-(t?Math.ceil(Math.log(t)/Math.LN10):
1)}function _(t){return t+""}function w(t,n){var e=Math.pow(10,3*Math.abs(8-n));return{scale:n>8?function(t)
{return t/e}:function(t){return t*e},symbol:t}}function S(t){return function(n){return 0>=n?0:n>=1?1:t(n)}}
function k(t){return function(n){return 1-t(1-n)}}function E(t){return function(n){return.5*(.5>n?t(2*n):2-
t(2-2*n))}}function A(t){return t*t}function N(t){return t*t*t}function T(t){if(0>=t)return 0;if(t>=1)return
1;var n=t*t,e=n*t;return 4*(.5>t?e:3*(t-n)+e-.75)}function q(t){return function(n){return Math.pow(n,t)}}
function C(t){return 1-Math.cos(t*Ni/2)}function z(t){return Math.pow(2,10*(t-1))}function D(t){return 1-
Math.sqrt(1-t*t)}function L(t,n){var e;return 2>arguments.length&&(n=.45),arguments.length?e=n/
(2*Ni)*Math.asin(1/t):(t=1,e=n/4),function(r){return 1+t*Math.pow(2,10*-r)*Math.sin(2*(r-e)*Ni/n)}}function F(t)
{return t||(t=1.70158),function(n){return n*n*((t+1)*n-t)}}function H(t){return 1/2.75>t?7.5625*t*t:2/2.75>t?
7.5625*(t-=1.5/2.75)*t+.75:2.5/2.75>t?7.5625*(t-=2.25/2.75)*t+.9375:7.5625*(t-=2.625/2.75)*t+.984375}function
j(){qi.event.stopPropagation(),qi.event.preventDefault()}function P(){for(var
t,n=qi.event;t=n.sourceEvent;)n=t;return n}function R(t){for(var n=new M,e=0,r=arguments.length;r>+
+e;)n[arguments[e]]=b(n);return n.of=function(e,r){return function(u){try{var
i=u.sourceEvent=qi.event;u.target=t,qi.event=u,n[u.type].apply(e,r)}finally{qi.event=i}}},n}function O(t){var
n=[t.a,t.b],e=[t.c,t.d],r=U(n),u=Y(n,e),i=U(I(e,n,-u))||
0;n[0]*e[1]<e[0]*n[1]&&(n[0]*=-1,n[1]*=-1,r*=-1,u*=-1),this.rotate=(r?Math.atan2(n[1],n[0]):Math.atan2(-
e[0],e[1]))*zi,this.translate=[t.e,t.f],this.scale=[r,i],this.skew=i?Math.atan2(u,i)*zi:0}function Y(t,n){return
t[0]*n[0]+t[1]*n[1]}function U(t){var n=Math.sqrt(Y(t,t));return n&&(t[0]/=n,t[1]/=n),n}function I(t,n,e){return
t[0]+=e*n[0],t[1]+=e*n[1],t}function V(t){return"transform"==t?qi.interpolateTransform:qi.interpolate}function
X(t,n){return n=n-(t=+t)?1/(n-t):0,function(e){return(e-t)*n}}function Z(t,n){return n=n-(t=+t)?1/(n-t):
0,function(e){return Math.max(0,Math.min(1,(e-t)*n))}}function B(){}function $(t,n,e){return new J(t,n,e)}
function J(t,n,e){this.r=t,this.g=n,this.b=e}function G(t){return
16>t?"0"+Math.max(0,t).toString(16):Math.min(255,t).toString(16)}function K(t,n,e){var r,u,i,a=0,o=0,c=0;if(r=/
([a-z]+)\((.*)\)/i.exec(t))switch(u=r[2].split(","),r[1]){case"hsl":return e(parseFloat(u[0]),parseFloat(u[1])/
100,parseFloat(u[2])/100);case"rgb":return n(nn(u[0]),nn(u[1]),nn(u[2]))}return(i=aa.get(t))?n(i.r,i.g,i.b):
(null!=t&&"#"===t.charAt(0)&&(4===t.length?(a=t.charAt(1),a+=a,o=t.charAt(2),o+=o,c=t.charAt(3),c+=c):
7===t.length&&(a=t.substring(1,3),o=t.substring(3,5),c=t.substring(5,7)),a=parseInt(a,16),o=parseInt(o,
16),c=parseInt(c,16)),n(a,o,c))}function W(t,n,e){var r,u,i=Math.min(t/=255,n/=255,e/
=255),a=Math.max(t,n,e),o=a-i,c=(a+i)/2;return o?(u=.5>c?o/(a+i):o/(2-a-i),r=t==a?(n-e)/o+(e>n?6:0):n==a?(e-t)/o
+2:(t-n)/o+4,r*=60):u=r=0,en(r,u,c)}function Q(t,n,e){t=tn(t),n=tn(n),e=tn(e);var r=pn((.4124564*t+.3575761*n+.
1804375*e)/fa),u=pn((.2126729*t+.7151522*n+.072175*e)/sa),i=pn((.0193339*t+.119192*n+.9503041*e)/ha);return
ln(116*u-16,500*(r-u),200*(u-i))}function tn(t){return.04045>=(t/=255)?t/12.92:Math.pow((t+.055)/1.055,2.4)}
function nn(t){var n=parseFloat(t);return"%"===t.charAt(t.length-1)?Math.round(2.55*n):n}function en(t,n,e)
```

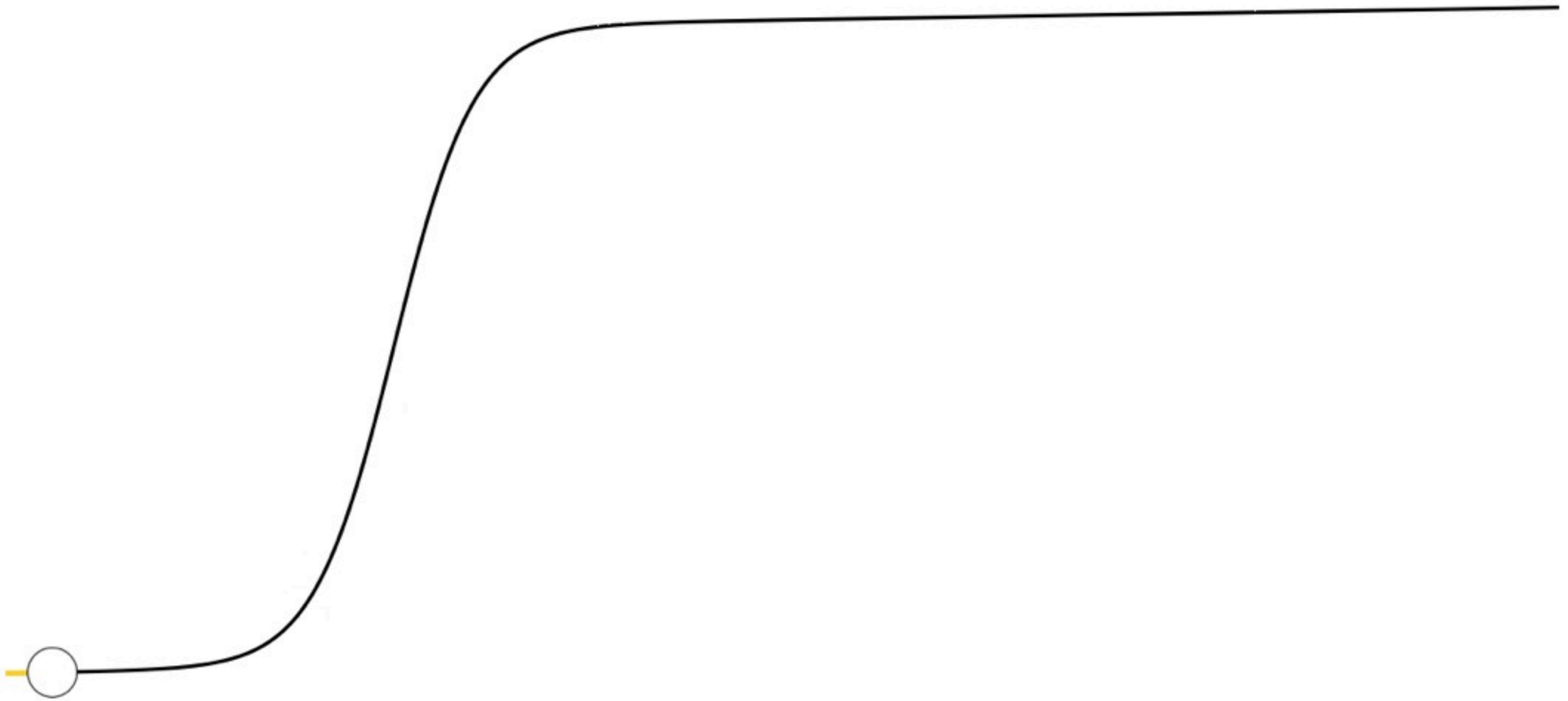
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>D3 Page Template</title>
  <script type="text/javascript" src="d3.v3.js"></script>
</head>
<body>
  <script type="text/javascript">
    // Your beautiful D3 code can go here
  </script>
</body>
</html>
```

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SVG	Scalable Vector Graphics
DOM	The Document Object Model

all of the above == web standards

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SVG	Scalable Vector Graphics
DOM	The Document Object Model

Learning D3 is a process of "learning the web"



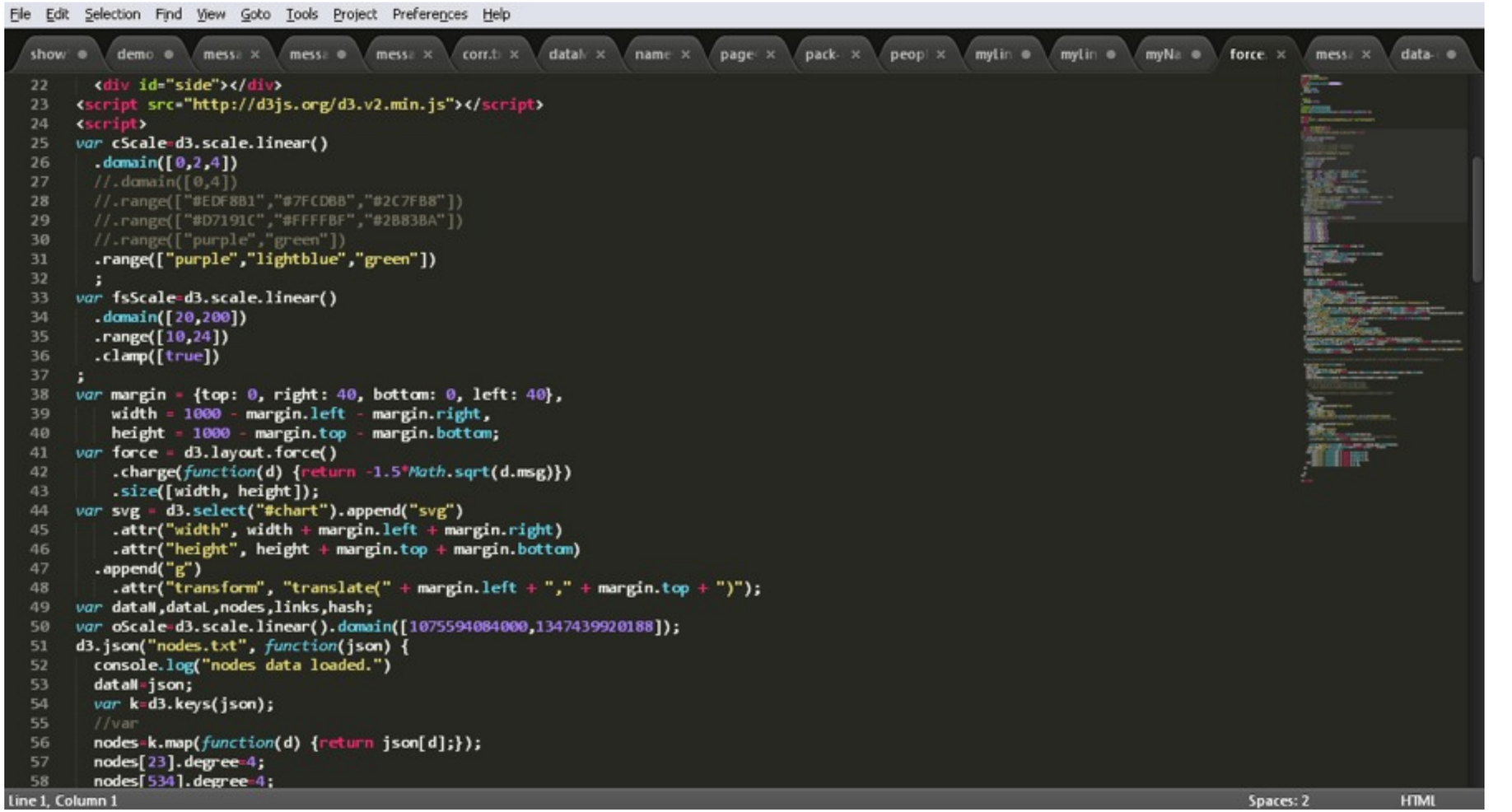
Your d3 learning curve

What you need

- A text editor,
- The d3 library,
- Files for your code,
- *Recommended:* a web server,
- A browser.

A text editor

- There are a few options out there: textMate, eclipse / aptana, sublime text 2…
- What you really need is an editor with **syntax highlighting**. Constructs with d3 can become very intricate.
- Personally, I like sublime text 2.


File Edit Selection Find View Goto Tools Project Preferences Help
show demo messa messa messa corr.t datal name page pack peopl mylin mylin myNa force messa data-

<script src="http://d3js.org/d3.v2.min.js"></script>
<script>
var cScale=d3.scale.linear()
.domain([0,2,4])
//.domain([0,4])
//.range(["#EDF8B1","#7FCDBB","#2C7FB8"])
//.range(["#D7191C","#FFFFBF","#2B83BA"])
//.range(["purple","green"])
.range(["purple","lightblue","green"])
;
var fsScale=d3.scale.linear()
.domain([20,200])
.range([10,24])
.clamp([true])
;
var margin = {top: 0, right: 40, bottom: 0, left: 40},
width = 1000 - margin.left - margin.right,
height = 1000 - margin.top - margin.bottom;
var force = d3.layout.force()
.charge(function(d) {return -1.5*Math.sqrt(d.msg)})
.size([width, height]);
var svg = d3.select("#chart").append("svg")
.attr("width", width + margin.left + margin.right)
.attr("height", height + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + "," + margin.top + ")");
var dataN,dataL,nodes,links,hash;
var oScale=d3.scale.linear().domain([1075594084000,1347439920188]);
d3.json("nodes.txt", function(json) {
console.log("nodes data loaded.")
dataN=json;
var k=d3.keys(json);
//var
nodes=k.map(function(d) {return json[d];});
nodes[23].degree=4;
nodes[534].degree=4;
Line 1, Column 1
Spaces: 2
HTML

Files you need

- The d3 library : get it at http://d3js.org
- Or link to it: http://d3js.org/d3. v3.min.js

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
            <title>My project</title>
           <script type="text/javascript" src="../d3.v3.js"></script>
           <link href="style.css" rel="stylesheet">
        </head>
        <body>
           <div id="chart"></div>
           <script type="text/javascript" src="script.js"></script>
       </body>
</html>
```

A template for d3

<!DOCTYPE html>

Start by specifying the doctype, to be in HTML5 mode (less suprises).

A template for d3

<!DOCTYPE html>
<html>
</html>

An HTML tag is not required, but makes things more legible for people.

A template for d3

```
<!DOCTYPE html>
<html>
      <head>
      </head>
      <body>
     </body>
</html>
```

Likewise, head and body tags are not required, but make things easier to read.

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
           <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
         </head>
         <body>
        </body>
</html>
```

It's better to specify a content type, this will allow you to use non-ascii characters with confidence.

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
          <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
          <title>My project</title>
        </head>
        <body>
       </body>
</html>
```

You may name your project here.

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
          <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
          <title>My project</title>
          <script type="text/javascript" src="http://d3js.org/d3.v3.js"></script>
        </head>
        <body>
       </body>
</html>
```

That's where you link to the d3 library. Here I am assuming it is in a folder one level up from the code. Alternatively, you can use http://d3js.org/d3.v2.min.js.

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
            <title>My project</title>
           <script type="text/javascript" src="../d3.v2.js"></script>
           <link href="style.css" rel="stylesheet">
        </head>
        <body>
       </body>
</html>
```

Optionally, you can link to a stylesheet like so. Or specify style inside a <style> element here.

```
<!DOCTYPE html>
<html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
            <title>My project</title>
           <script type="text/javascript" src="http://d3js.org/d3.v3.js"></script>
           <link href="style.css" rel="stylesheet">
        </head>
        <body>
            <div id="chart"></div>
       </body>
</html>
```

Inside the body, we create a element which will hold the vis.

```
<!DOCTYPE html>
<html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
            <title>My project</title>
           <script type="text/javascript" src="http://d3js.org/d3.v3.js"></script>
           <link href="style.css" rel="stylesheet">
        </head>
        <body>
           <div id="chart"></div>
           <script type="text/javascript" src="script.js"></script>
       </body>
</html>
```

Finally, we link to a script file containing our actual javascript code.
Alternatively, we may write our code here inside a <script> element.

A template for d3

```
<!DOCTYPE html>
<html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
            <title>My project</title>
           <script type="text/javascript" src="http://d3js.org/d3.v3.js"></script>
           <link href="style.css" rel="stylesheet">
        </head>
        <body>
           <div id="chart"></div>
           <script type="text/javascript" src="script.js"></script>
       </body>
</html>
```

Now let's look at a sample js file.

```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
    .attr("width",w).attr("height",h);
svg
    .append("text")
    .text("hello world!").attr("x",100).attr("y",100);
```

Now let's look at a sample js file.

var w=960,h=500;

Simple variables to size the vis.

Those numbers are chosen because they work well with Mike Bostock's http://bl.ocks.org, a simple viewer for code examples hosted on GitHub Gist.

Now let's look at a sample js file.

```
var w=960,h=500;
var svg=d3.select("#chart")
```

Now we are going to create an SVG container. It will be a child of the div named #chart, which we created earlier.

Now let's look at a sample js file.

```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
```

This creates the svg element per se.

Now let's look at a sample js file.

```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
     .attr("width",w).attr("height",h);
```

And this last line gives an explicit width and height to the svg element. This is desired in Firefox (in chrome/safari, the svg just resizes as needed) and generally more proper.

Now let's look at a sample js file.

```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
    .attr("width",w).attr("height",h);
svg
    .append("text")
```

Now that we have an SVG container, we can just add any kind of SVG element to it. So let's start with text.

Now let's look at a sample js file.

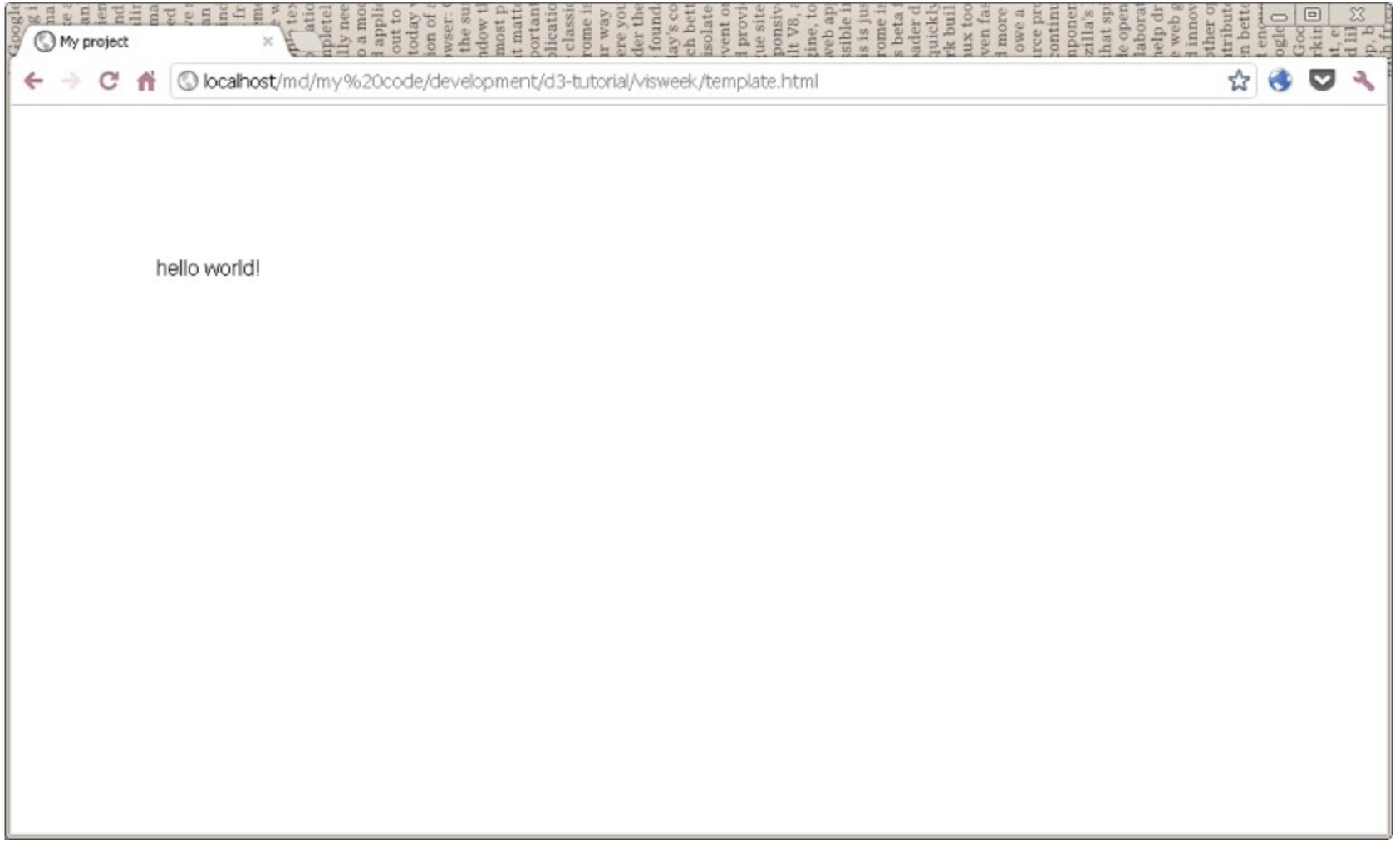
```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
    .attr("width",w).attr("height",h);
svg
    .append("text")
    .text("hello world!").attr("x",100).attr("y",100);
```

This last line specifies characteristics of the element we've just added.

A sample js file.

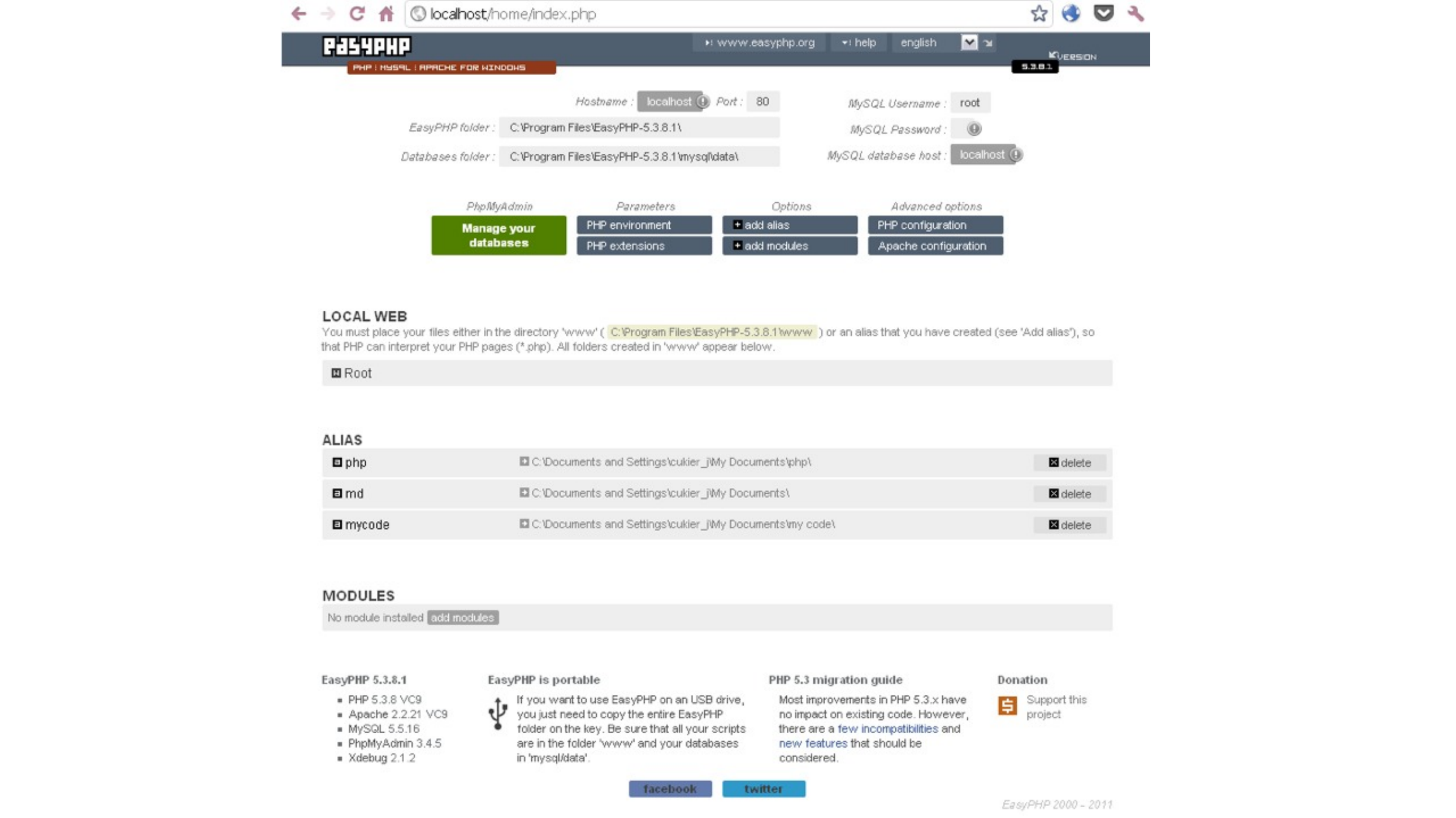
```
var w=960,h=500;
var svg=d3.select("#chart")
    .append("svg")
    .attr("width",w).attr("height",h);
svg
    .append("text")
    .text("hello world!").attr("x",100).attr("y",100);
```

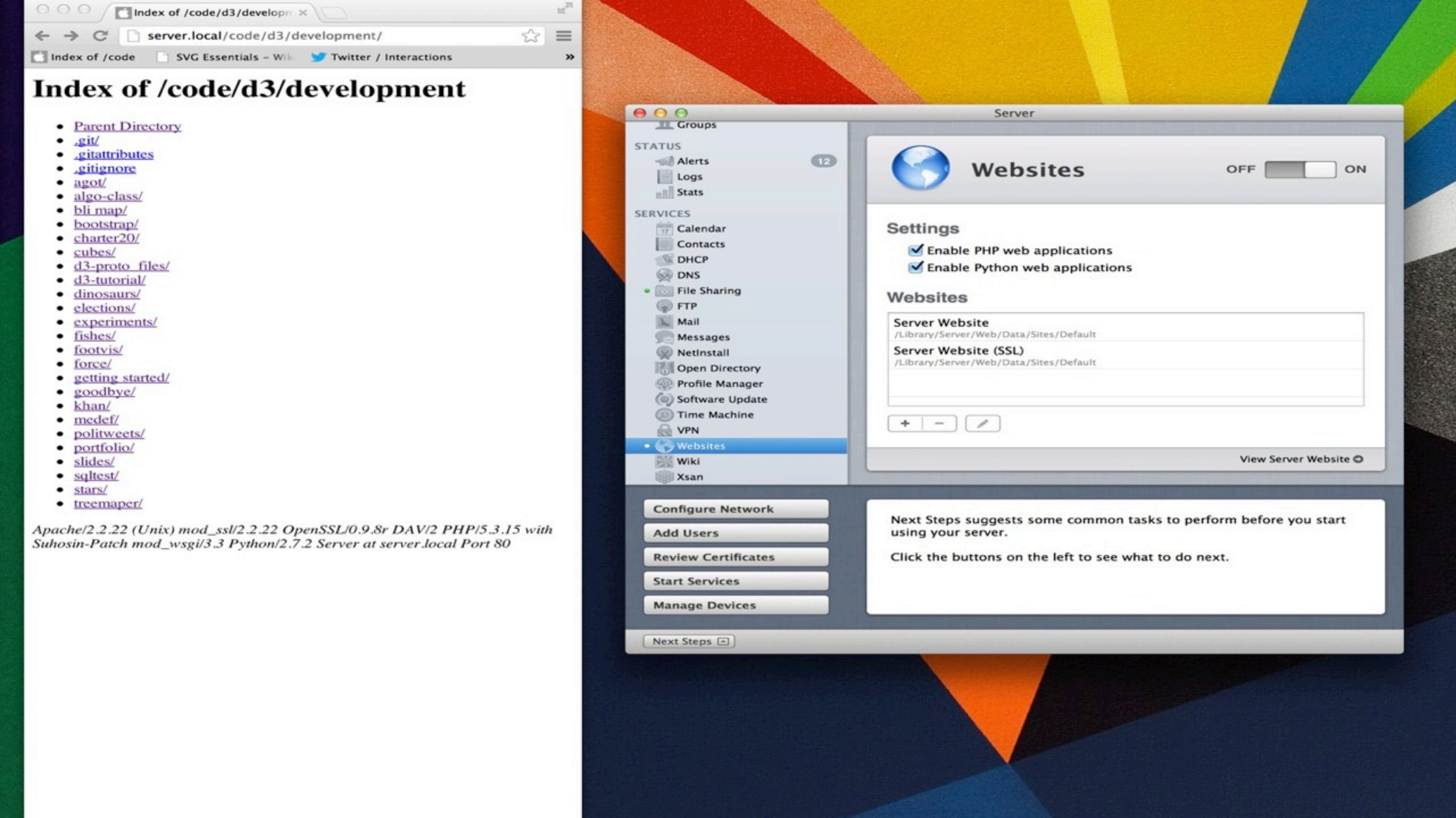
Lo and behold:



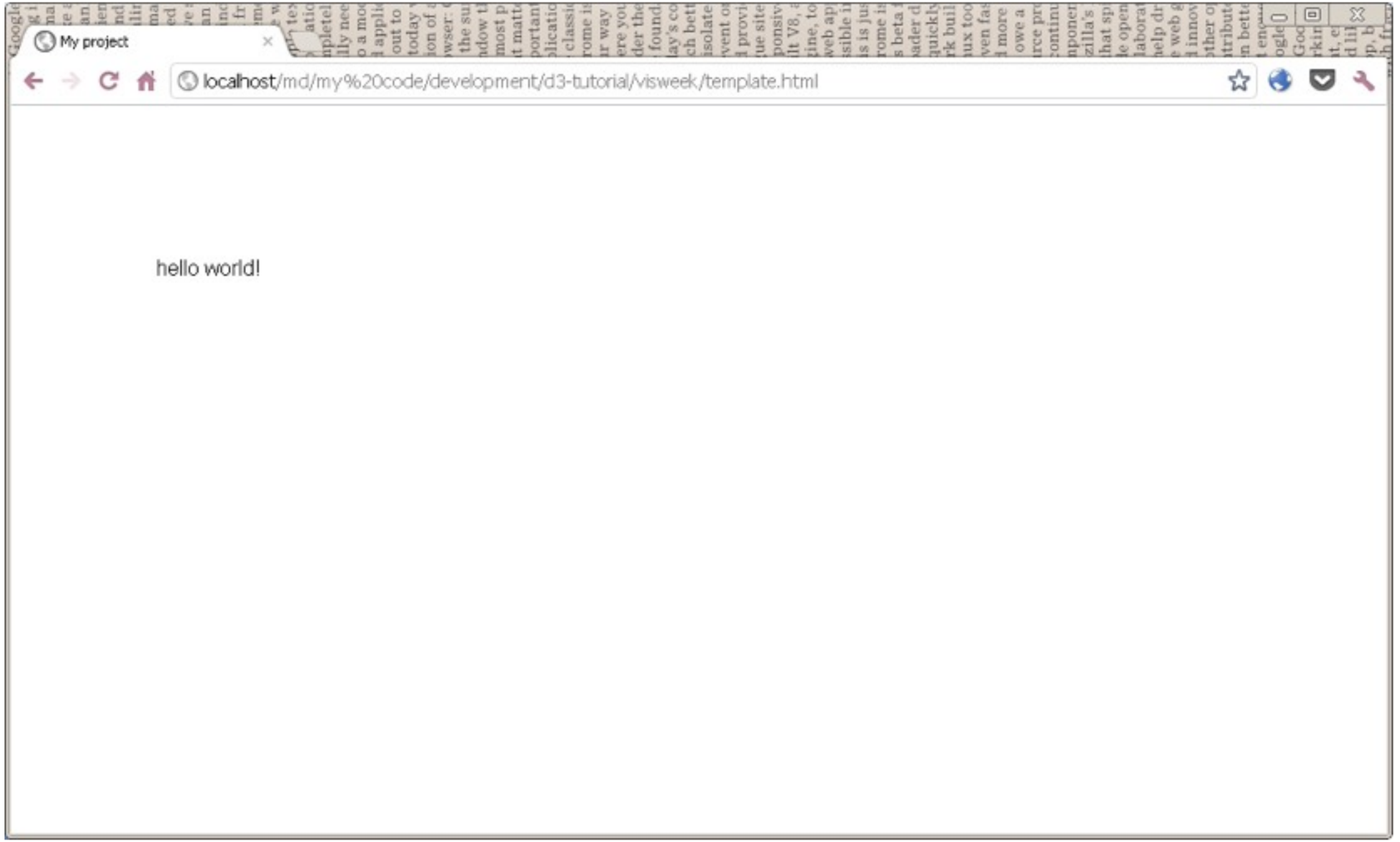
A web server

- You can view most d3 visualizations locally, simply by opening an html file in a browser.
- But if your visualization is reading data from files or from a database (XMLHttpRequest), then you need to publish it on a web server to test it.
- There are many options: EasyPHP (windows), Mac OS X Server, MAMP (Mac OS X)

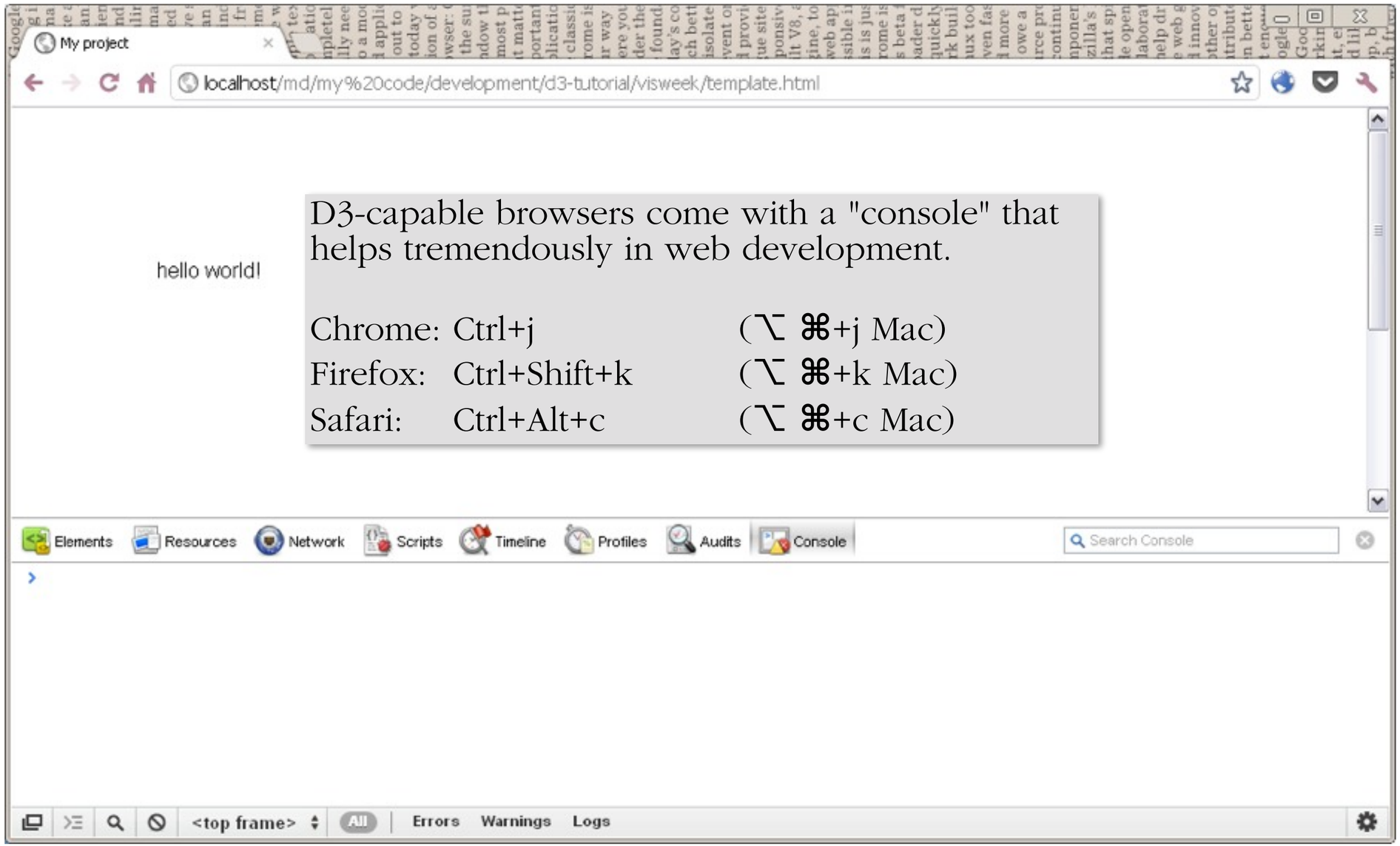

localhost/home/index.php
EASYPHP
PHP : MYSQL : APACHE FOR WINDOWS
www.easyphp.org
help
english
VERSION
5.3.8.1
Hostname : localhost Port : 80
EasyPHP folder : C:\Program Files\EasyPHP-5.3.8.1\
Databases folder : C:\Program Files\EasyPHP-5.3.8.1\mysql\data\
MySQL Username : root
MySQL Password :
MySQL database host : localhost
PhpMyAdmin
Manage your databases
Parameters
PHP environment
PHP extensions
Options
add alias
add modules
Advanced options
PHP configuration
Apache configuration
LOCAL WEB
You must place your files either in the directory 'www' (C:\Program Files\EasyPHP-5.3.8.1\www) or an alias that you have created (see 'Add alias'), so that PHP can interpret your PHP pages (*.php). All folders created in 'www' appear below.
Root
ALIAS
php C:\Documents and Settings\cukier_j\My Documents\php\ delete
md C:\Documents and Settings\cukier_j\My Documents\ delete
mycode C:\Documents and Settings\cukier_j\My Documents\my code\ delete
MODULES
No module installed add modules
EasyPHP 5.3.8.1
PHP 5.3.8 VC9
Apache 2.2.21 VC9
MySQL 5.5.16
PhpMyAdmin 3.4.5
Xdebug 2.1.2
EasyPHP is portable
If you want to use EasyPHP on an USB drive, you just need to copy the entire EasyPHP folder on the key. Be sure that all your scripts are in the folder 'www' and your databases in 'mysql/data'.
PHP 5.3 migration guide
Most improvements in PHP 5.3.x have no impact on existing code. However, there are a few incompatibilities and new features that should be considered.
Donation
Support this project
facebook
twitter
EasyPHP 2000 - 2011


Index of /code/d3/developr
server.local/code/d3/development/
Index of /code
SVG Essentials – Wik
Twitter / Interactions
Index of /code/d3/development
Parent Directory
.git/
.gitattributes
.gitignore
agot/
algo-class/
bli map/
bootstrap/
charter20/
cubes/
d3-proto_files/
d3-tutorial/
dinosaurs/
elections/
experiments/
fishes/
footvis/
force/
getting started/
goodbye/
khan/
medef/
politweets/
portfolio/
slides/
sqltest/
stars/
treemaper/
Apache/2.2.22 (Unix) mod_ssl/2.2.22 OpenSSL/0.9.8r DAV/2 PHP/5.3.15 with Suhosin-Patch mod_wsgi/3.3 Python/2.7.2 Server at server.local Port 80
Server
Groups
STATUS
Alerts
12
Logs
Stats
SERVICES
Calendar
Contacts
DHCP
DNS
File Sharing
FTP
Mail
Messages
NetInstall
Open Directory
Profile Manager
Software Update
Time Machine
VPN
Websites
Wiki
Xsan
Websites
OFF
ON
Settings
Enable PHP web applications
Enable Python web applications
Websites
Server Website
/Library/Server/Web/Data/Sites/Default
Server Website (SSL)
/Library/Server/Web/Data/Sites/Default
View Server Website
Configure Network
Add Users
Review Certificates
Start Services
Manage Devices
Next Steps suggests some common tasks to perform before you start using your server.
Click the buttons on the left to see what to do next.
Next Steps

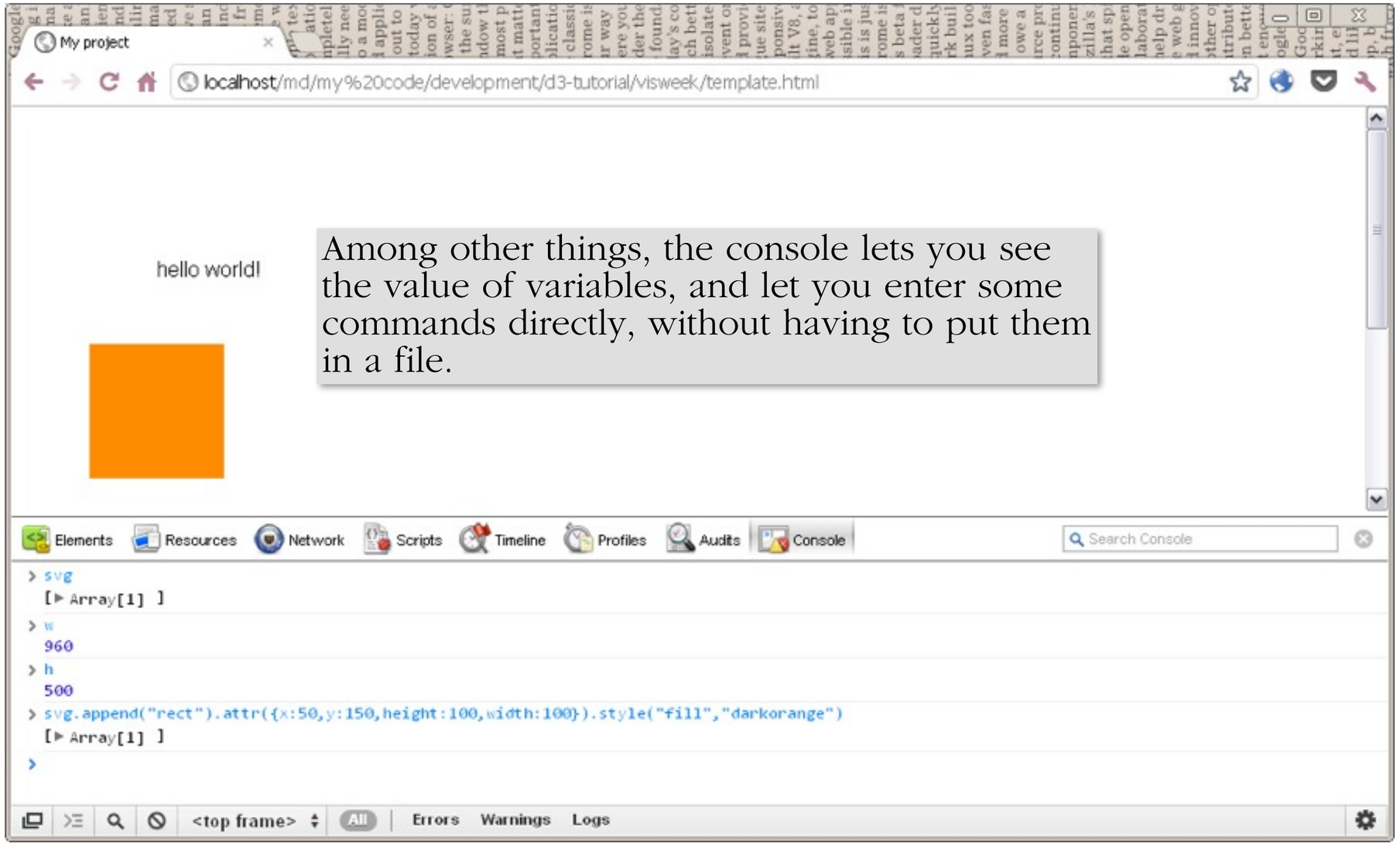
Finally, a browser

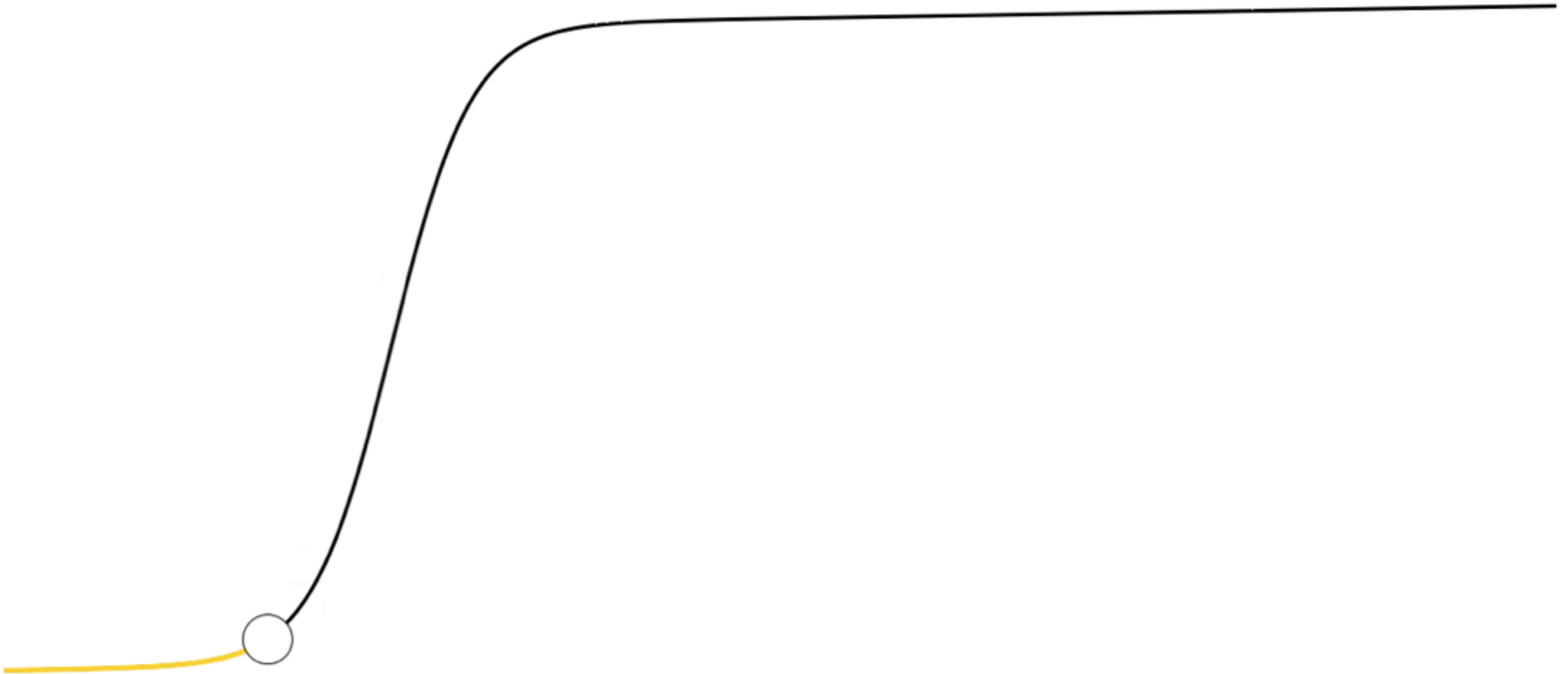


The console


My project
localhost/md/my%20code/development/d3-tutorial/visweek/template.html
hello world!
D3-capable browsers come with a "console" that helps tremendously in web development.
Chrome: Ctrl+j (⌥ ⌘+j Mac)
Firefox: Ctrl+Shift+k (⌥ ⌘+k Mac)
Safari: Ctrl+Alt+c (⌥ ⌘+c Mac)
Elements
Resources
Network
Scripts
Timeline
Profiles
Audits
Console
Search Console
<top frame>
All
Errors
Warnings
Logs

The console


My project
localhost/md/my%20code/development/d3-tutorial/visweek/template.html
hello world!
Among other things, the console lets you see the value of variables, and let you enter some commands directly, without having to put them in a file.
Elements
Resources
Network
Scripts
Timeline
Profiles
Audits
Console
Search Console
> svg
[▶ Array[1]]
> w
960
> h
500
> svg.append("rect").attr({x:50,y:150,height:100,width:100}).style("fill","darkorange")
[▶ Array[1]]
>
<top frame>
All
Errors
Warnings
Logs



Your d3 learning curve

Selecting elements

Exercise: Create this web page by typing D3 code into the console.

h1

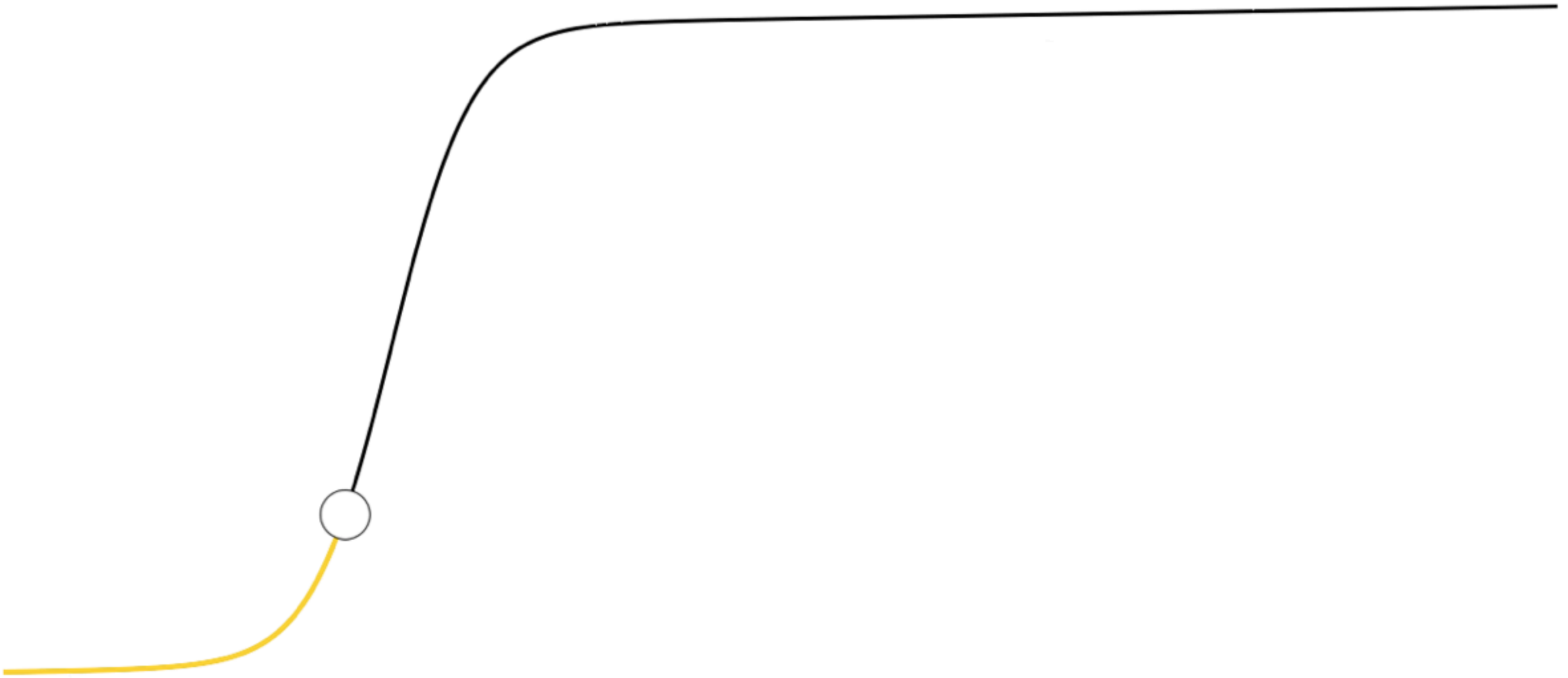
p

a

Strata Tutorial

D3 can be used to generate new DOM elements.

Get it from d3js.org!



Your d3 learning curve

Data joins with d3

Strata d3 tutorial

Where's the data?

So far we've been adding elements one by one.
Let's put the data in **data visualization!**

Introducting selectAll

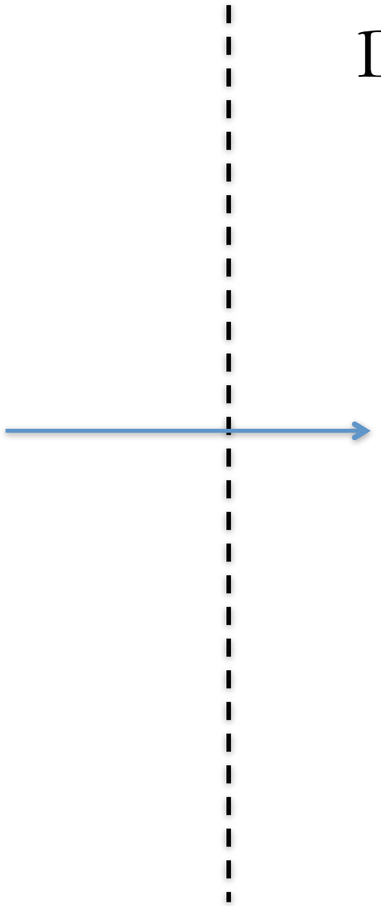
selectAll allows you to select all elements that correspond to a condition, and manipulate them all at once.

```
d3.selectAll("p").style("font-weight","bold");
```

Here's how it works:

All existing elements

Do stuff

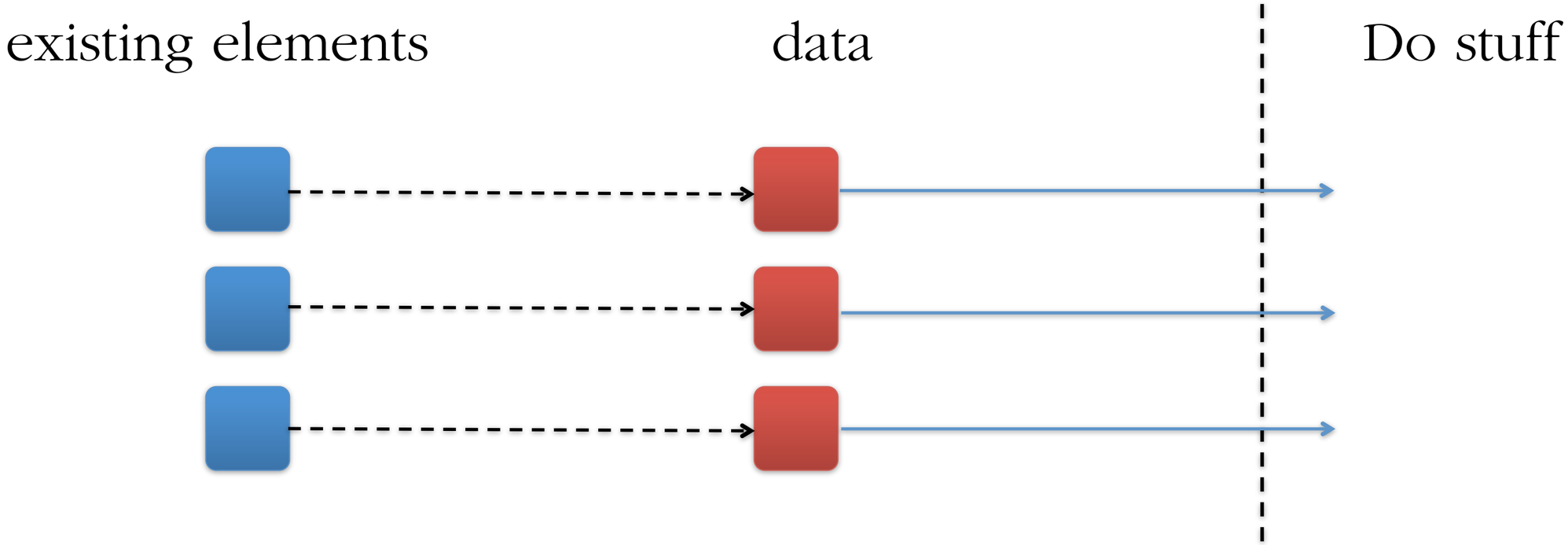


Values based on data

Instead of asking d3 to do the same thing unconditionally, we can ask it to update certain characteristics of the items based on data.

```
var fs= ["10px","20px","30px"];
d3.selectAll("p")
.data(fs)
.style("font-size",function(d) {return d;})
```

What just happened?



Side note: who is d?

Here I wrote:
.style("font-size",function(d) {return d;})

What is d?

Here, I'm assigning to the "font-size" style a value which is not static, but dynamic.

To compute that value, we retrieve it from the data using **functions**.

The first argument of these functions is the data item. The *name* of that argument is arbitrary, *d* is a convention.

Here, we just return the value we've read, but inside the function there can be any kind of transformation.

Side note: can I use existing functions instead of retyping them each time?

YES!!

For instance, String(123) converts a number into a string.
Conveniently, it also converts a string into a string.
In most cases, String(d) is equivalent to function(d) {return d;}

So instead of .style("font-size",function(d) {return d;})
We can write: .style("font-size",String)

Creating **new elements** from data

With selectAll, we've seen we can manipulate *existing elements*.

With selectAll then data, we've seen that we can manipulate *existing elements* dynamically, using data.

Now what if there are no existing elements?

Introducing enter

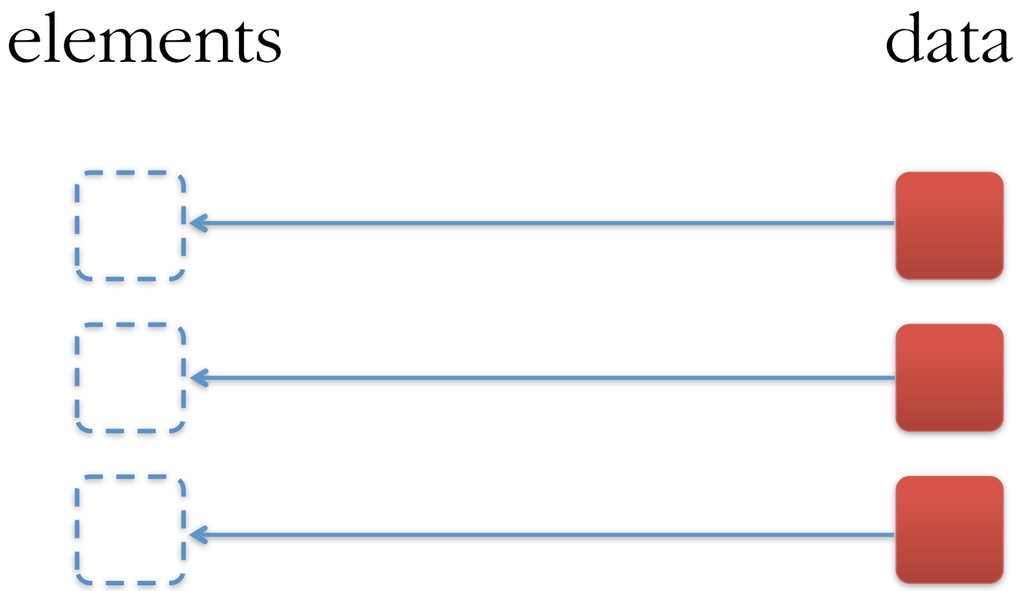
If you find yourself in this situation…

existing elements

data

Introducing enter

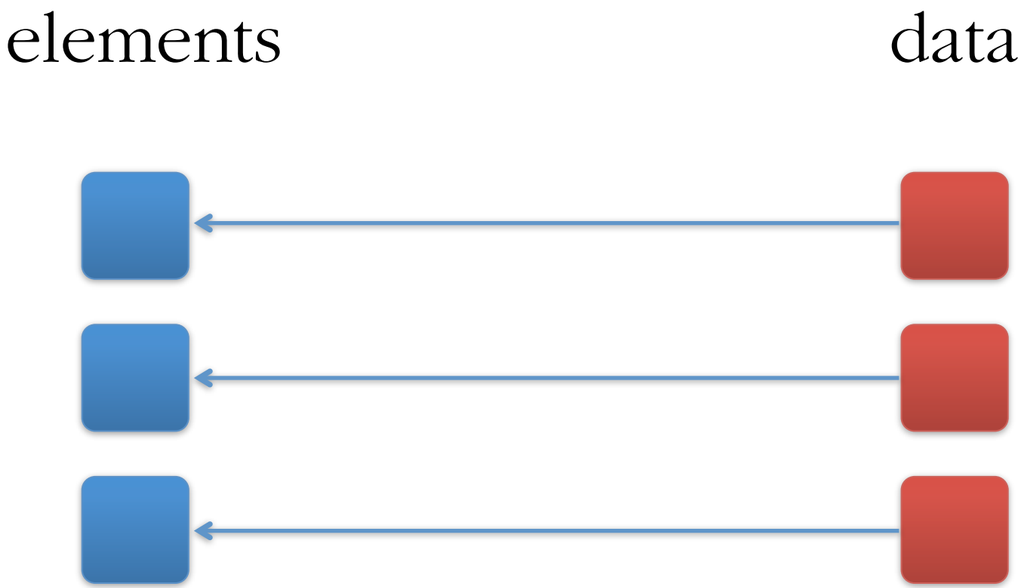
d3.select("body").selectAll("p").data(data)**.enter()**



will create "pods" for future elements

Introducing enter

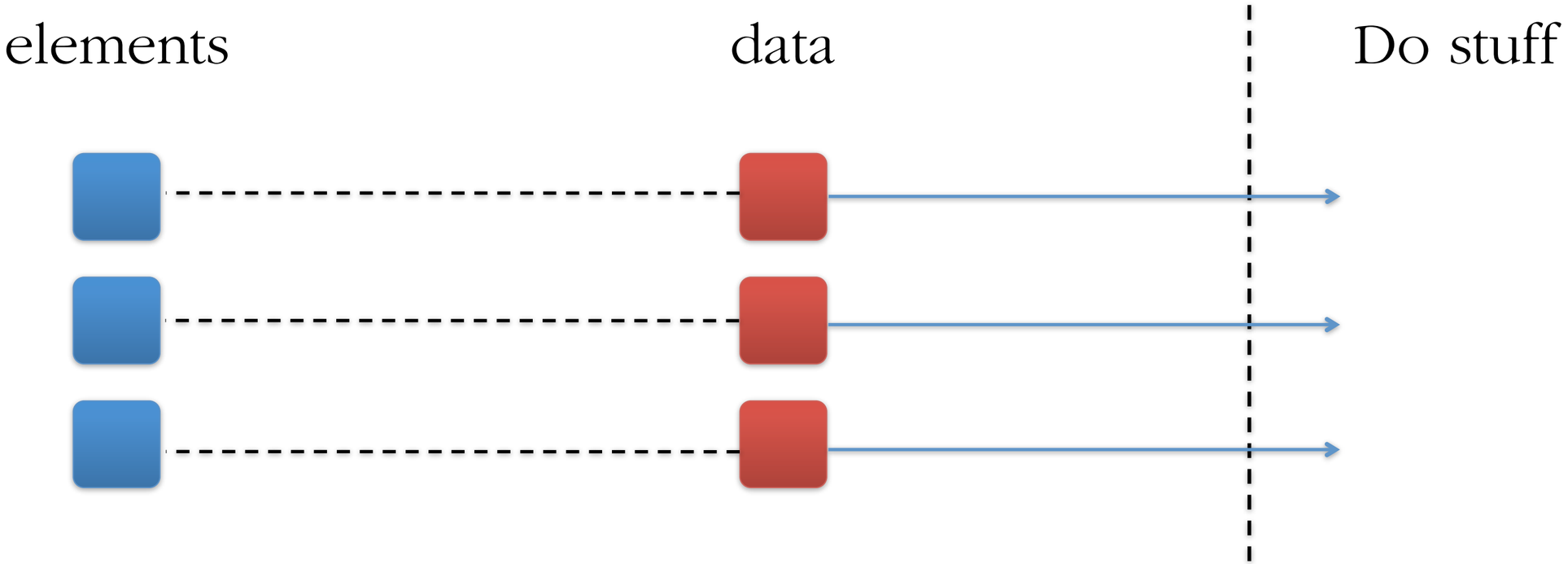
....selectAll("p").data(data).enter().**append("p")**



Then append will create as many elements as needed, as opposed to just one.

Working with new elements

… .html(function(d) …).style(…)



And after this, you can chain methods that will update characteristics of the elements as above.

Side note: why select before selectAll?

In previous examples (before enter) we could write directly: d3.selectAll("p").

This will **not work** when creating new elements.

Why? Because new elements have to be created **somewhere**!

So, for this construct to work, you have to select a container first, ie

d3**.select("body")**.selectAll("p").data(…).enter()…

Side note: what happens if the data changes?

So you created a bunch of elements dynamically, using data.
Then, for some reason, the data changes.
What happens to your elements?

Side note: what happens if the data changes?

So you created a bunch of elements dynamically, using data.
Then, for some reason, the data changes.
What happens to your elements?
Nothing, unless you also change their attributes.

(BTW – this is different from protovis, the ancestor of d3)

How do I remove elements?

The remove() method can be attached to any selection.

d3.selectAll("p").remove()

Effectively deletes all paragraphs found in the document.

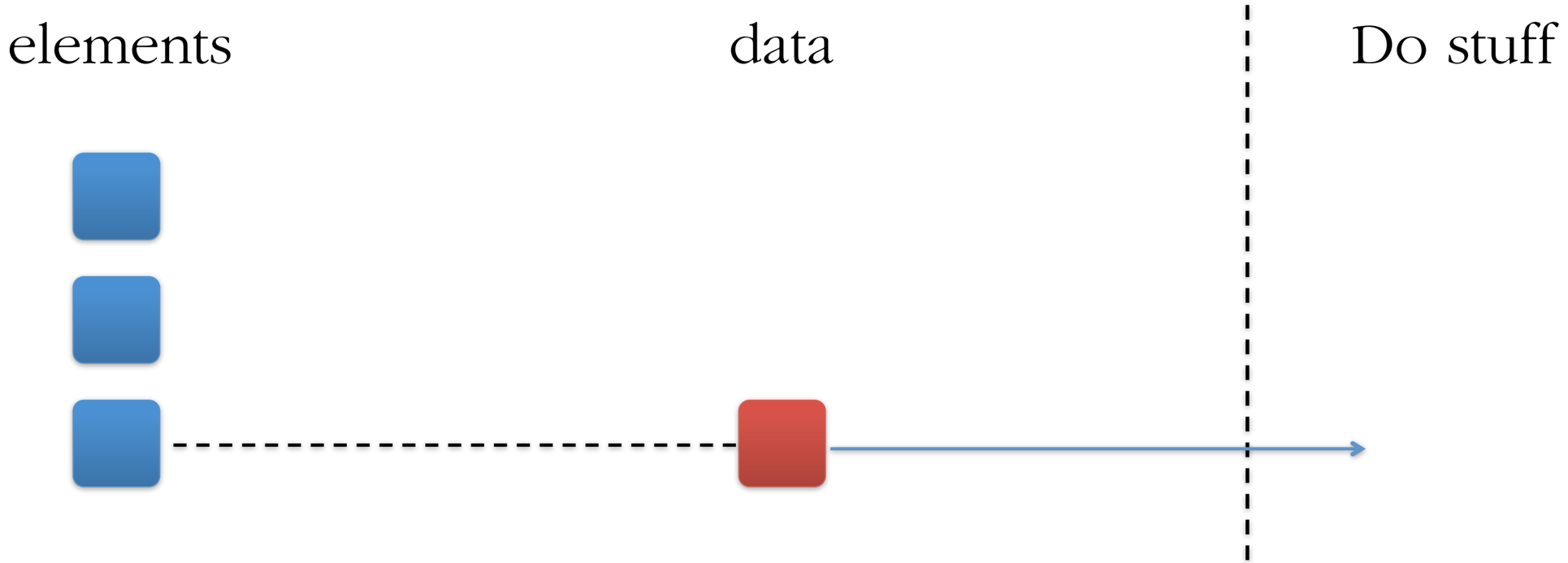
How do I remove **some** elements?

OK so let's suppose my data changes and I have fewer data points than I have created elements.

What happens if I want to manipulate my elements?

d3.selectAll("p").data(["hello world"]).html(String);

Working with a smaller dataset



Only the first element changes. The other two are left untouched.

How do I remove **some** elements?

In order to capture the elements which are no longer matched by a data point, we can use the method exit():

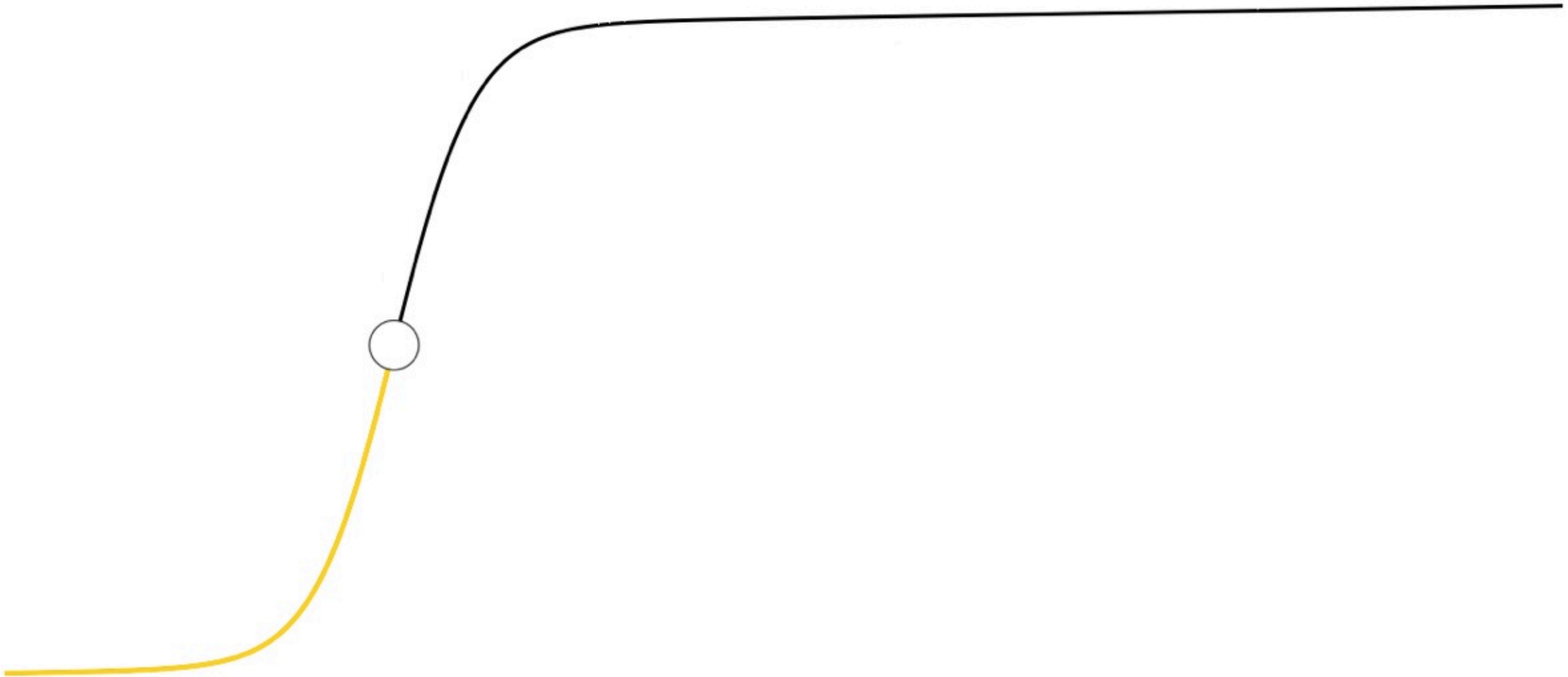
```
d3.selectAll("p").data(["hello world"]).exit()
// do stuff to those, often .remove()
;
```

Exercise: Create four span elements with the following text and colors.

span

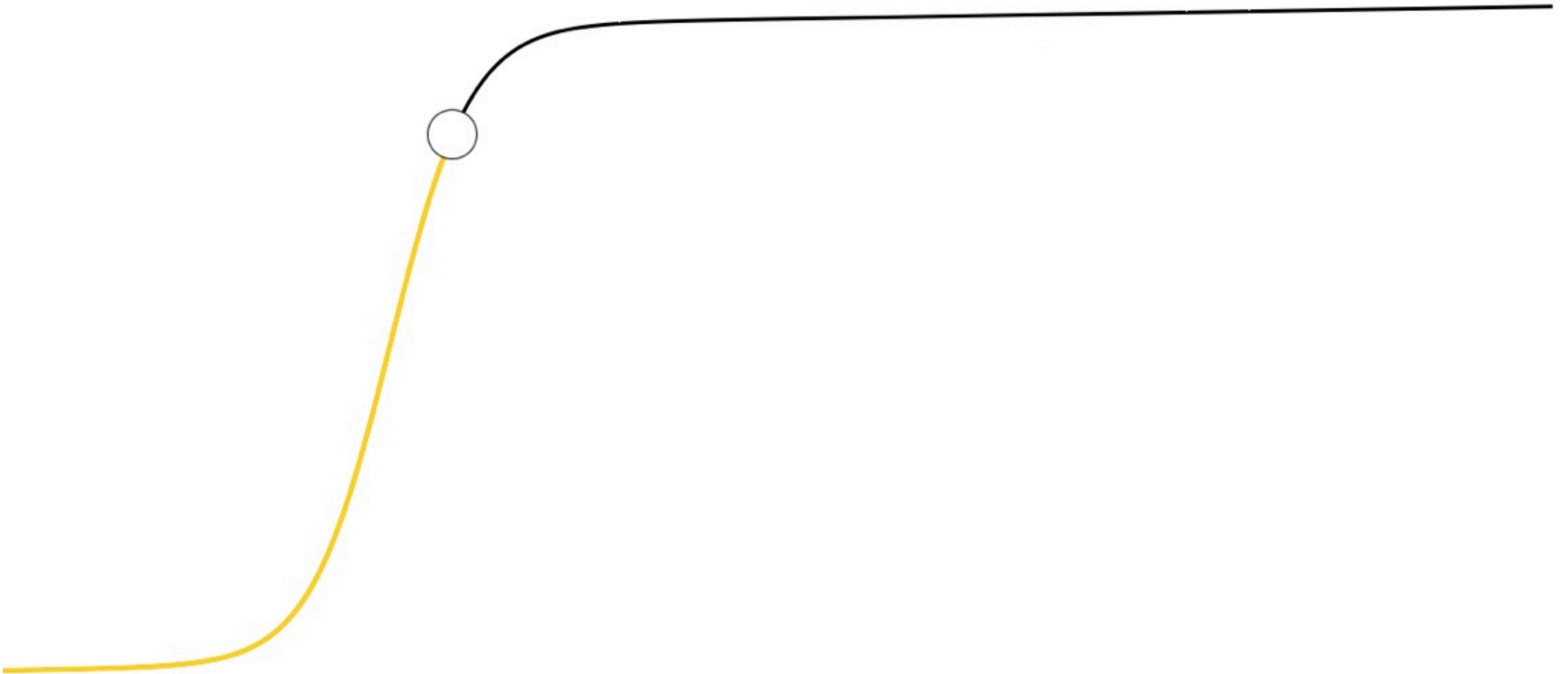
darkmagenta teal rosybrown midnightblue

```
HINT: var colors = [“darkmagenta ”,
                    “teal “,
                    “rosybrown “,
                    “midnightblue “]
```



Your d3 learning curve

SVG



Your d3 learning curve

Scales with d3

Strata d3 tutorial

Calculations from data

```
.attr("y", function(d) { return h - (d * 4); })
.attr("height", function(d) { return d * 4;})
```

What if the size of the chart changes?
Will that work if the shape of the data changes?

Introducing scales

Scales are a family of d3 methods for
simple algebraic transformations.
Here's one

```
var myScale=d3.scale.linear().domain([0,25])
    .range([0,100]);
myScale(0) // 0
myScale(25) // 100
myScale(10) // 40
```

Wait!

```
var myScale=d3.scale.linear().domain([0,25])
    .range([0,100]);
```

Isn't that the same as just **multiplying by 4?**
And if so, why would I care?

Advantages of a scale

- it's very easy to change its parameters.
- If you are changing the size of the elements… without a scale you'd have to change every possible instance of the hard coded calculations.

.attr("y", function(d) { return **h - (d * 4)**; })

.attr("height", function(d) { return **d * 5**;})

This is very error-prone.

Using scales can lead to nice, compact yet legible code

```
var y=d3.scale.linear().range([0,100]).domain([0,25]);


....
    .attr("y",y)
```

Using scales can lead to nice, compact yet legible code

```
var y=d3.scale.linear().range([0,100]).domain([0,25]);


….
    .attr("y",y)
```

There are several other types of scales:

– log
– sqrt
– power

There are lots of other niceties that come with scales

```
y.domain(d3.extent(dataset)) // computes bounds of the scale automatically
y.domain([0,d3.max(dataset)]) // another way of determining the domain automatically
y.clamp([true]) // values outside the bounds of the domain get the min or max value of the range.
y.invert() // the mapping in the reverse direction.
```

So…

What's the sweet spot between using scales (which means having to write scales in full once) and writing out functions quickly that do equivalent things?

So…

My recommendation:

Always.
Use.
Scales.

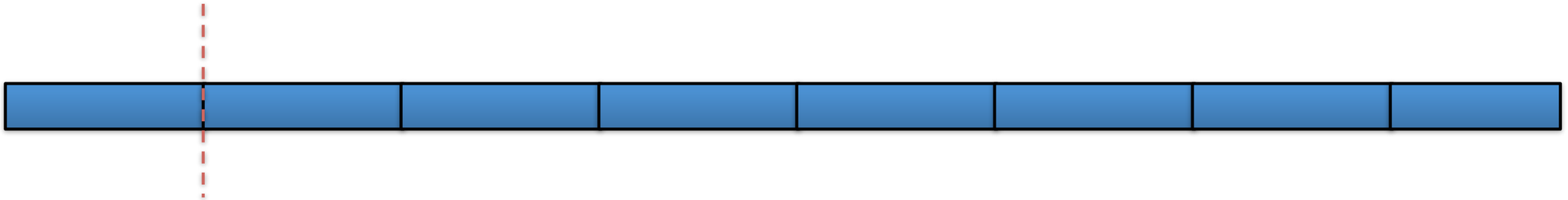
Ordinal scales

So far we have seen quantitative scales, which transform a **number** into another **number.**

But there are also *ordinal scales* which turn associate a list of items with a value.

rangeBands

```
myScale=d3.scale.ordinal()
.domain(["Monday","Tuesday","Wednesday","Thursday","Friday"
,"Saturday","Sunday"])
.rangeBands([0,100])

myScale("Tuesday")
```



rangePoints

```
myScale=d3.scale.ordinal()
.domain(["Monday","Tuesday","Wednesday","Thursday","Friday"
,"Saturday","Sunday"])
.rangePoints([0,100])

myScale("Tuesday")
```

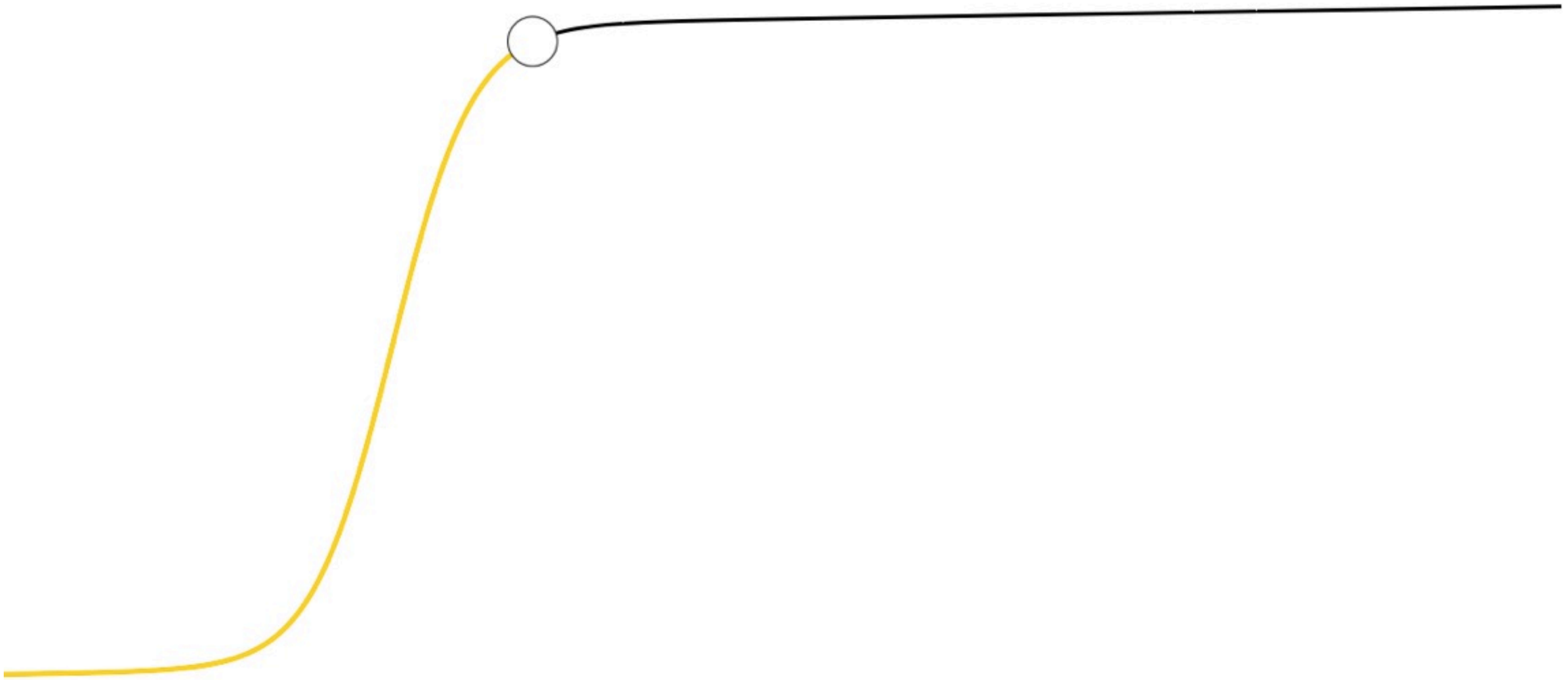
Interesting ordinal scales: color palettes!

In d3, color palettes are really ordinal scales, since they associate discrete values with values.

There are a few built in ones:

d3.scale.category10()

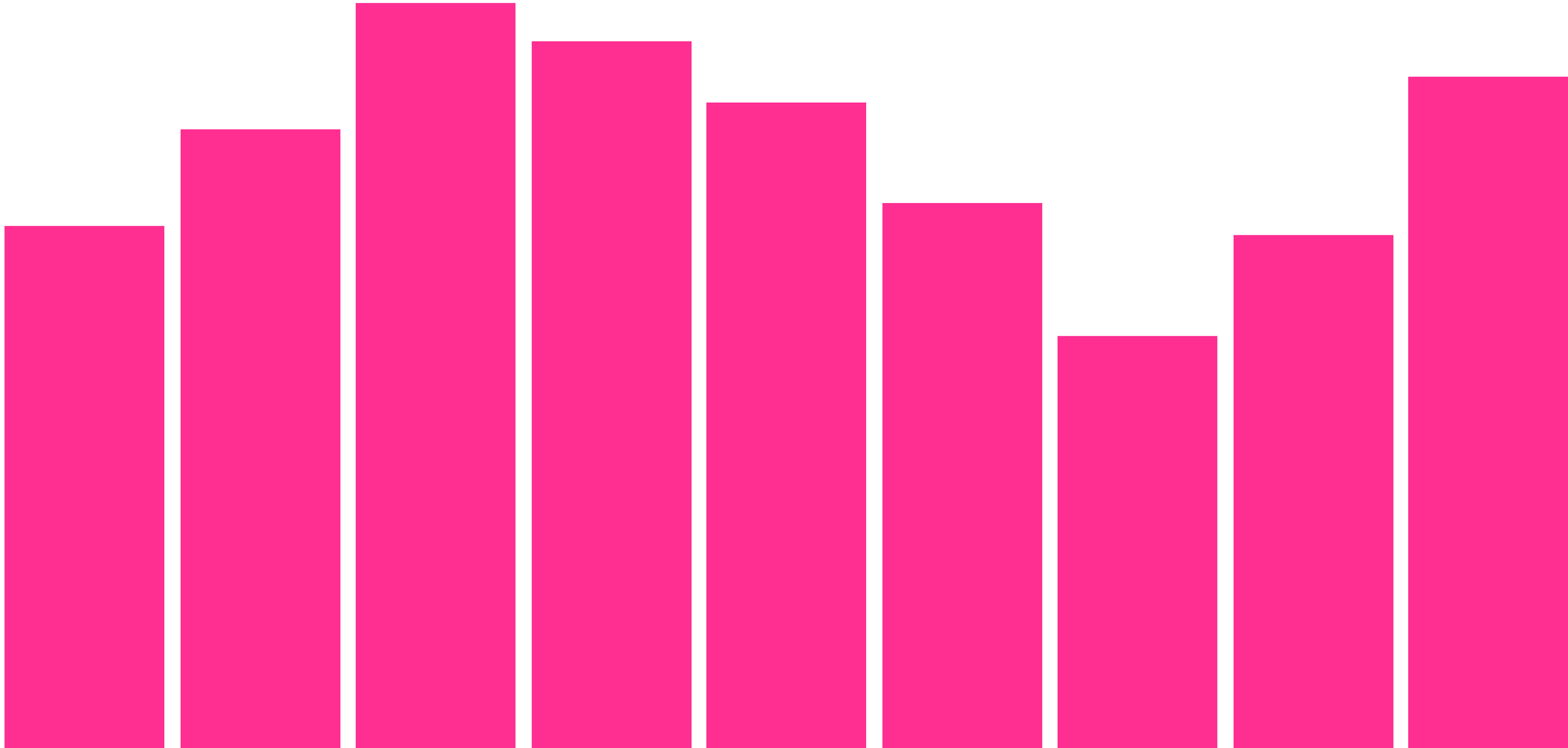
//d3.scale.category10()("a") //



Your d3 learning curve

Transitions

// Transitions and motion



Interaction with d3

Visweek d3 workshop

Two broad types of interaction with d3.

- Forms

And it doesn't have to be a full-fledged form: controls like drop-down menus, tick boxes, sliders etc.

- Interaction on elements of the chart proper

SVG or HTML elements: clicking, moving the cursor in or out, etc.

Good news!

While they may look different, they both really work the same.

And it’s not super complicated.

Here's an example

```
var w=960,h=500,flag=0;
var svg=d3.select("#chart").append("svg").attr("width",w).attr("height",h);
var myRect=svg
    .append( "rect").attr({x:100,y:100,width:100,height:100})
    .style("fill","steelblue");

myRect.on("click",function() {
    flag=1-flag;
    myRect.style("fill", flag?"darkorange":"steelblue");
})
```

Here's an example

var w=960,h=500,**flag=0**;

Note this flag variable initially set to 0.

```
var w=960,h=500,flag=0;
var svg=d3.select("#chart").append("svg").attr("width",w).attr("height",h);
var myRect=svg
    .append( "rect").attr({x:100,y:100,width:100,height:100})
    .style("fill","steelblue");
```

(Here we used a shorthand notation to avoid typing 4 .attr methods. Nothing to do with interaction but hey)

```
var w=960,h=500,flag=0;
var svg=d3.select("#chart").append("svg").attr("width",w).attr("height",h);
var myRect=svg
    .append( "rect").attr({x:100,y:100,width:100,height:100})
    .style("fill","steelblue");

myRect.on("click",function() {

})
```

That's where the action is.
on method, an event, a function.

```
var w=960,h=500,flag=0;
var svg=d3.select("#chart").append("svg").attr("width",w).attr("height",h);
var myRect=svg
    .append( "rect").attr({x:100,y:100,width:100,height:100})
    .style("fill","steelblue");

myRect.on("click",function() {
    flag=1-flag;
    myRect.style("fill", flag?"darkorange":"steelblue");
})
```

And now for the win:
we just toggle the value of flag (0 becomes 1 and vice versa)
then we style our rectangle according to that value : orange if flag is 1, else blue.

on + event + function

That's the general gist of it.

There are few events you should know.

"**click**" is the workhorse of events. Click anything (a rectangle, text, a shape) and things happen.

"**mouseover**", "**mouseout**" are other good ones.

Great for highlighting stuff and all.

"**change**" is great for forms (the value of the form changed).

Going further: events on groups

By setting an event listener on a "g" element, it will be triggered by any interaction on any element of that group.

Going further: events and data

Can the function access the underlying data of the element? Of course!

If the element has data tied to it, you can do

```
myElement.on("click",function(d) {
   // … operations on data …
})
```

Going further: playing with forms

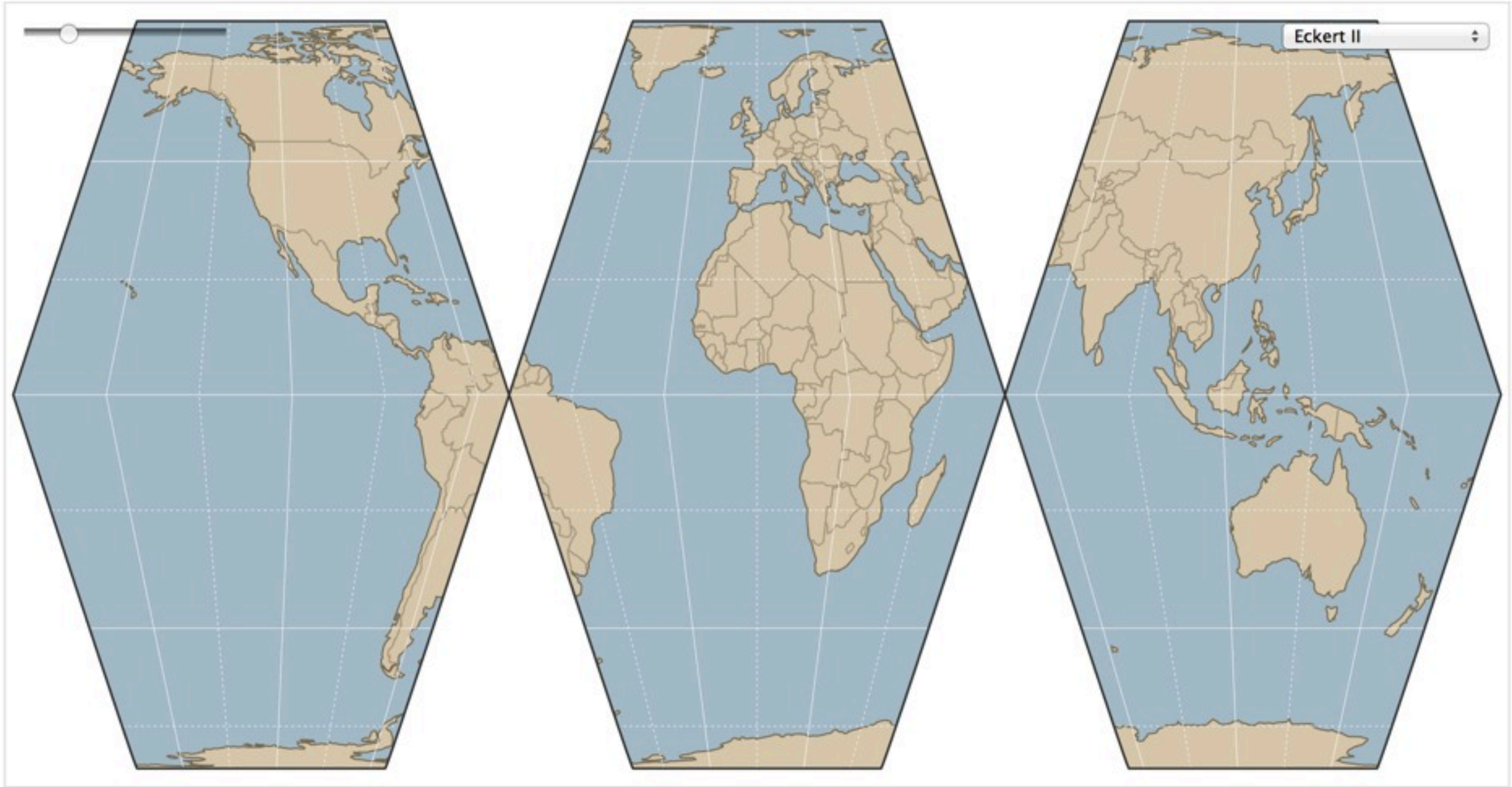
Usually the point of form controls is to access the value selected by the user. So you'll often have something like:

```
myControl.on("change",function() {
   doStuff(this.value);
})
```

this.value will store the value of the control.

Interrupted Projection Transitions

September 28, 2012



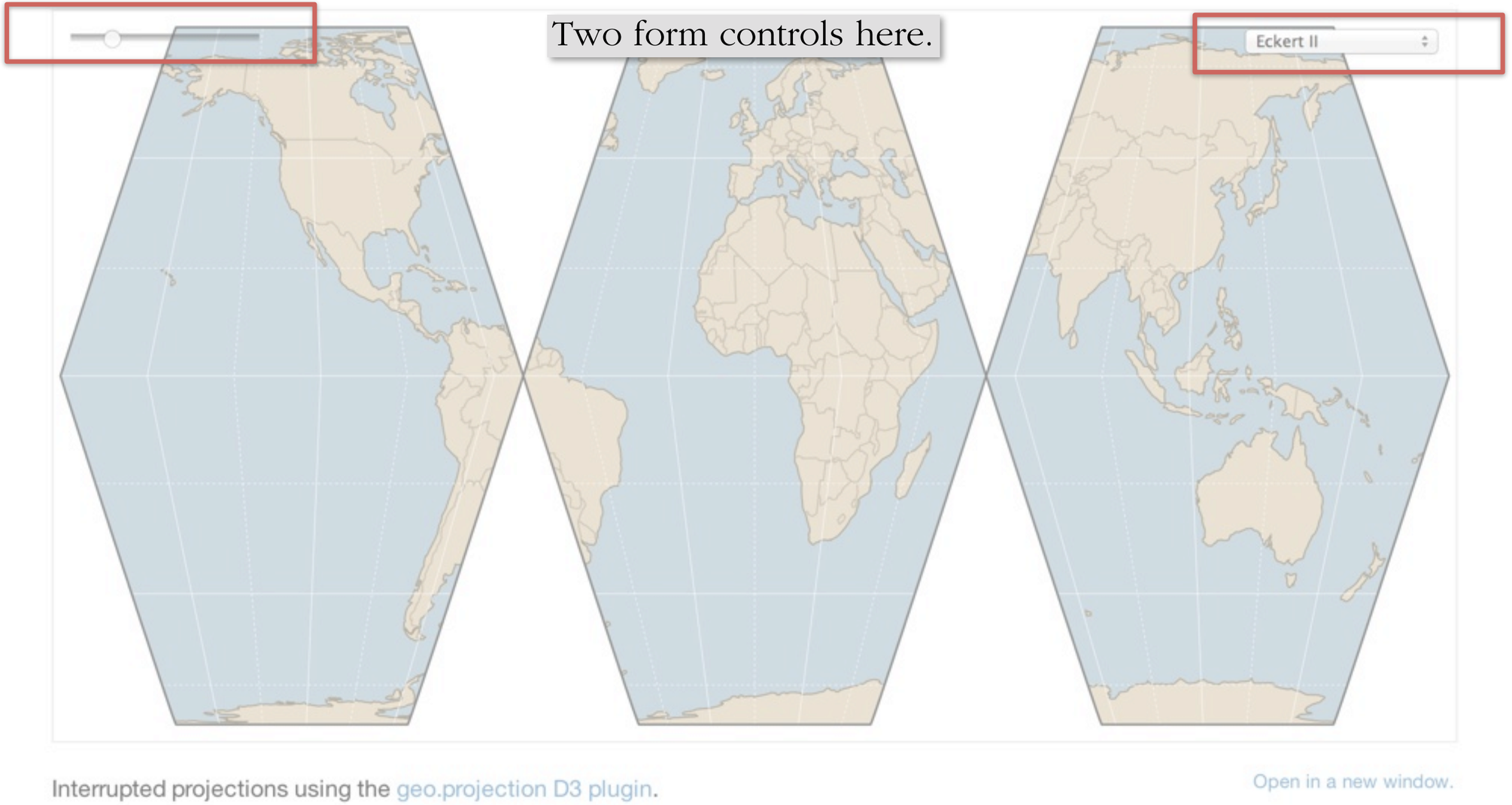
Interrupted projections using the geo.projection D3 plugin.

Open in a new window.

http://bl.ocks.org/3739100

Interrupted Projection Transitions

September 28, 2012



Interrupted projections using the geo.projection D3 plugin.

Open in a new window.

http://bl.ocks.org/3739100

Let's see how it looks like in the code

```
<body>
<select id="projection-menu"></select>
<input type="range" id="gores" step="1" min="1" max="12" value="3">
```

The two controls are created there:
- drop-down (select)
- slider (input with type "range")

Let's see how it looks like in the code

```
d3.select("#gores").on("change", function() {
        gores();
        …
});
…
function gores() {
  var n = +d3.select("#gores").property("value");
   …
}
```

When the slider value changes, the "change" event is triggered and so the gores() function is called.

Then, this function does stuff depending on the position of the slider, which is obtained thusly: by looking up the property "value" of the slider.

Let's see how it looks like in the code

```
menu.selectAll("option")
     .data(options)
   .enter().append("option")
     .text(function(d) { return d.name; });

var menu = d3.select("#projection-menu")
     .on("change", change);

function change() {
   …
   update(options[this.selectedIndex]);
}
```

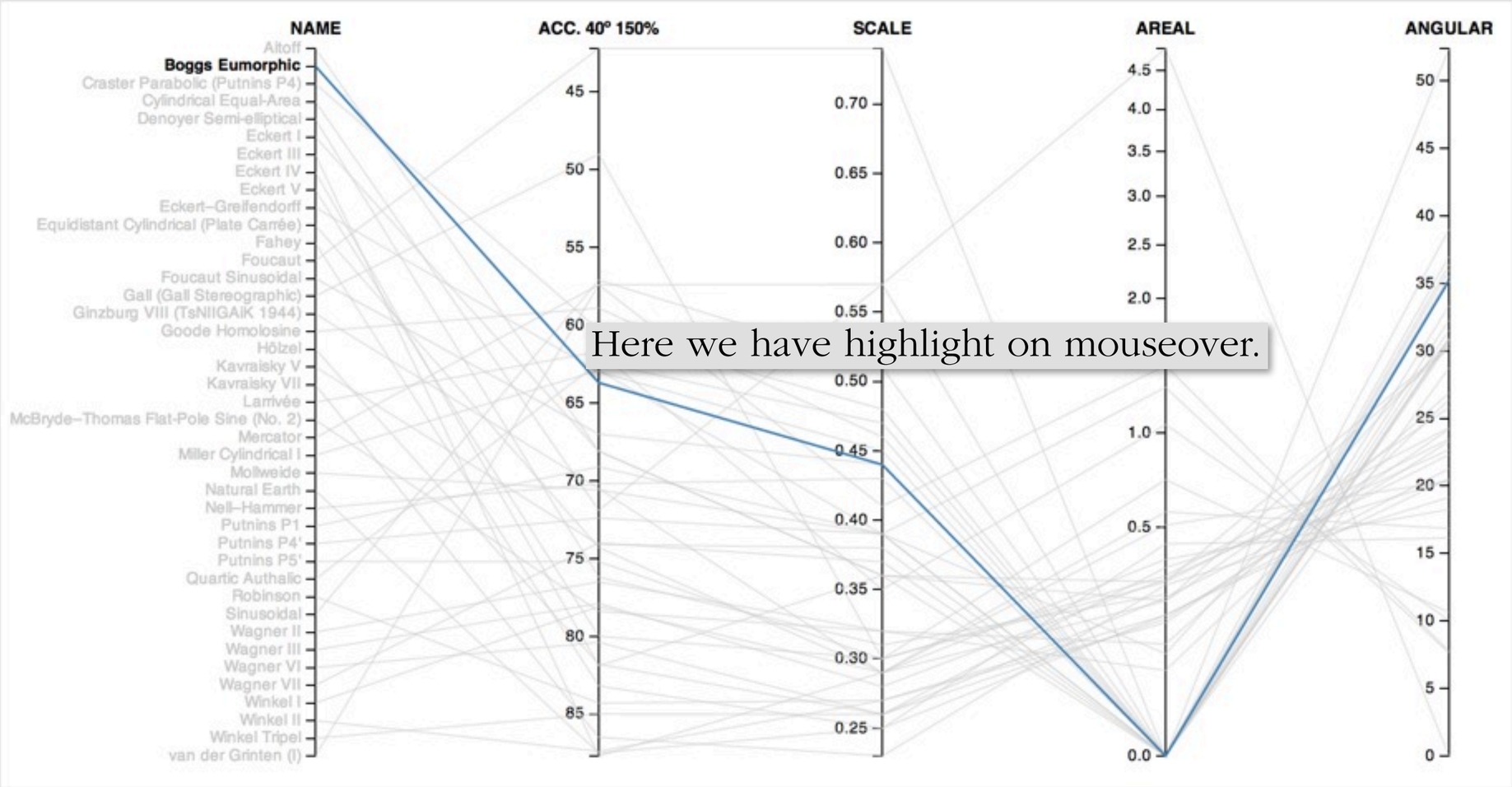
Here, the menu is going to be populated with data.

Then, just as before, a change event will trigger a function : change.

This function will call another one based on which item is selected in the drop down menu.

Map Projection Distortions

September 12, 2012



A comparison of 41 map projections by four different types of distortion. Lower is better. Data transcribed from the Natural Earth Projection.

Open in a new window.

http://bl.ocks.org/3709000

Let's see how it looks like in the code

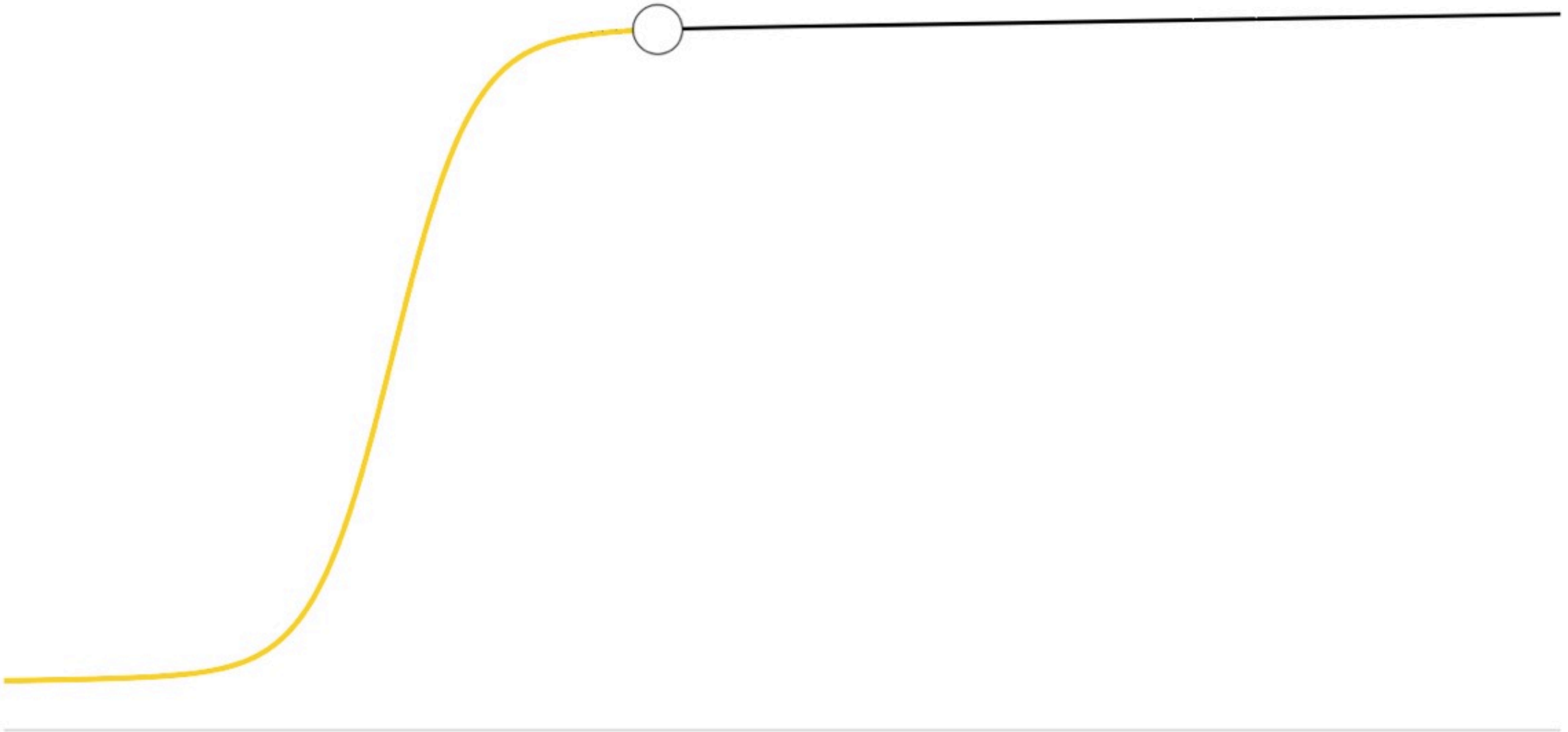
```
var projection = svg.selectAll(".axis text,.background path,.foreground path")
      .on("mouseover", mouseover)
      .on("mouseout", mouseout);

 function mouseover(d) {
   svg.classed("active", true);
   projection.classed("inactive", function(p) { return p !== d; });
   projection.filter(function(p) { return p === d; }).each(moveToFront);
 }

 function mouseout(d) {
   svg.classed("active", false);
   projection.classed("inactive", false);
 }

 function moveToFront() {
   this.parentNode.appendChild(this);
 }
```

This is our on - event – function

This function is called upon mouseover.
It uses the underlying value of the line (d).

The formatting is done by assigning a CSS class.

And this function is called upon mouseout, it essentially resets the other one.



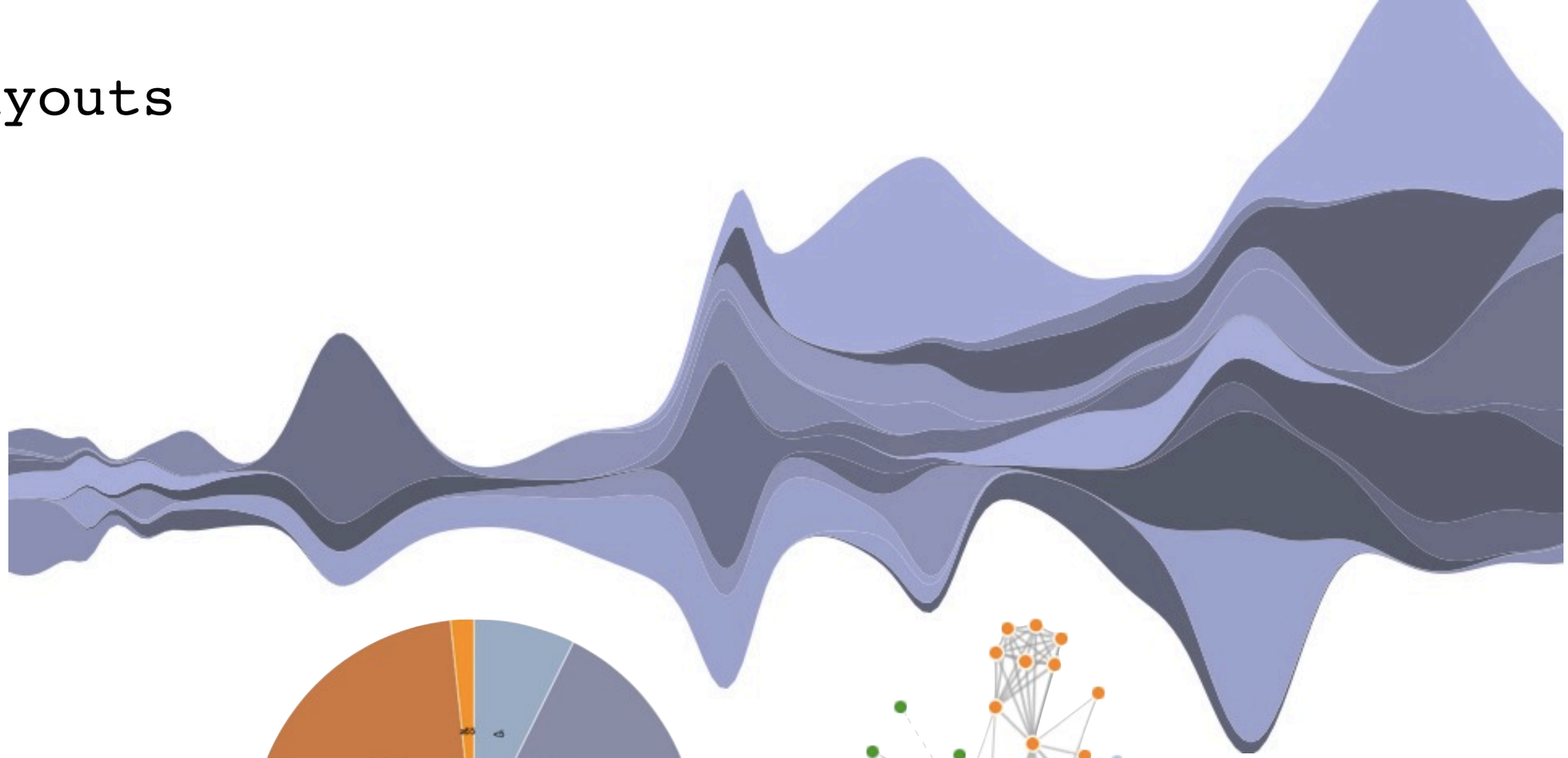
Your d3 learning curve

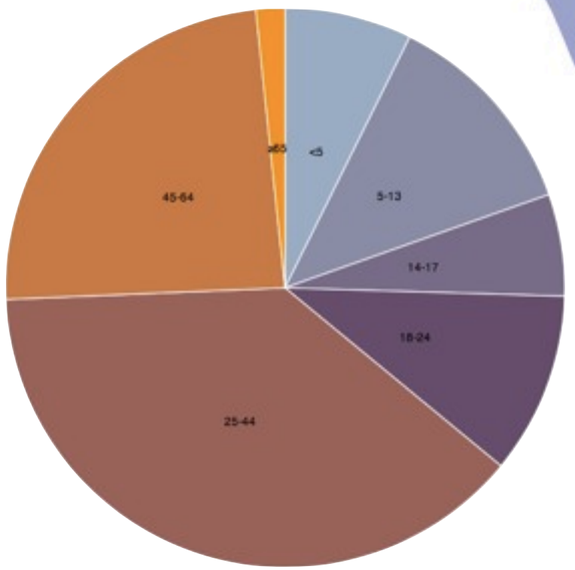
Questions?

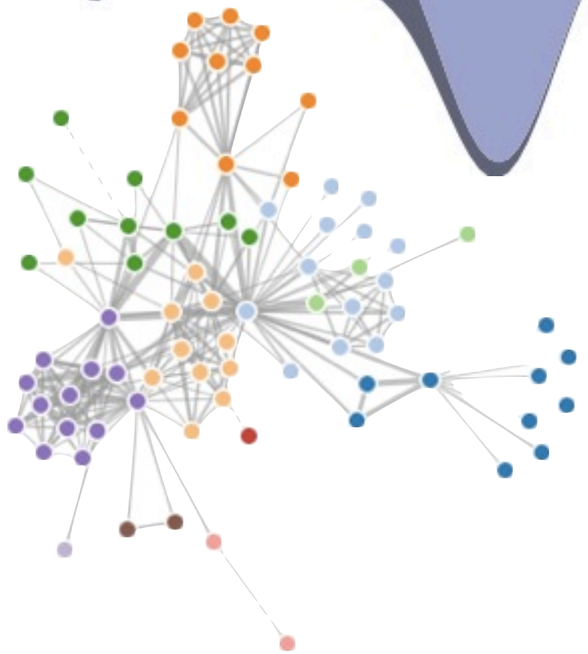
What's next?

```
// Generate axes

var axis = d3.svg.axis()
                 .scale(scale);

svg.append("g")
   .call(axis);
```

// Layouts





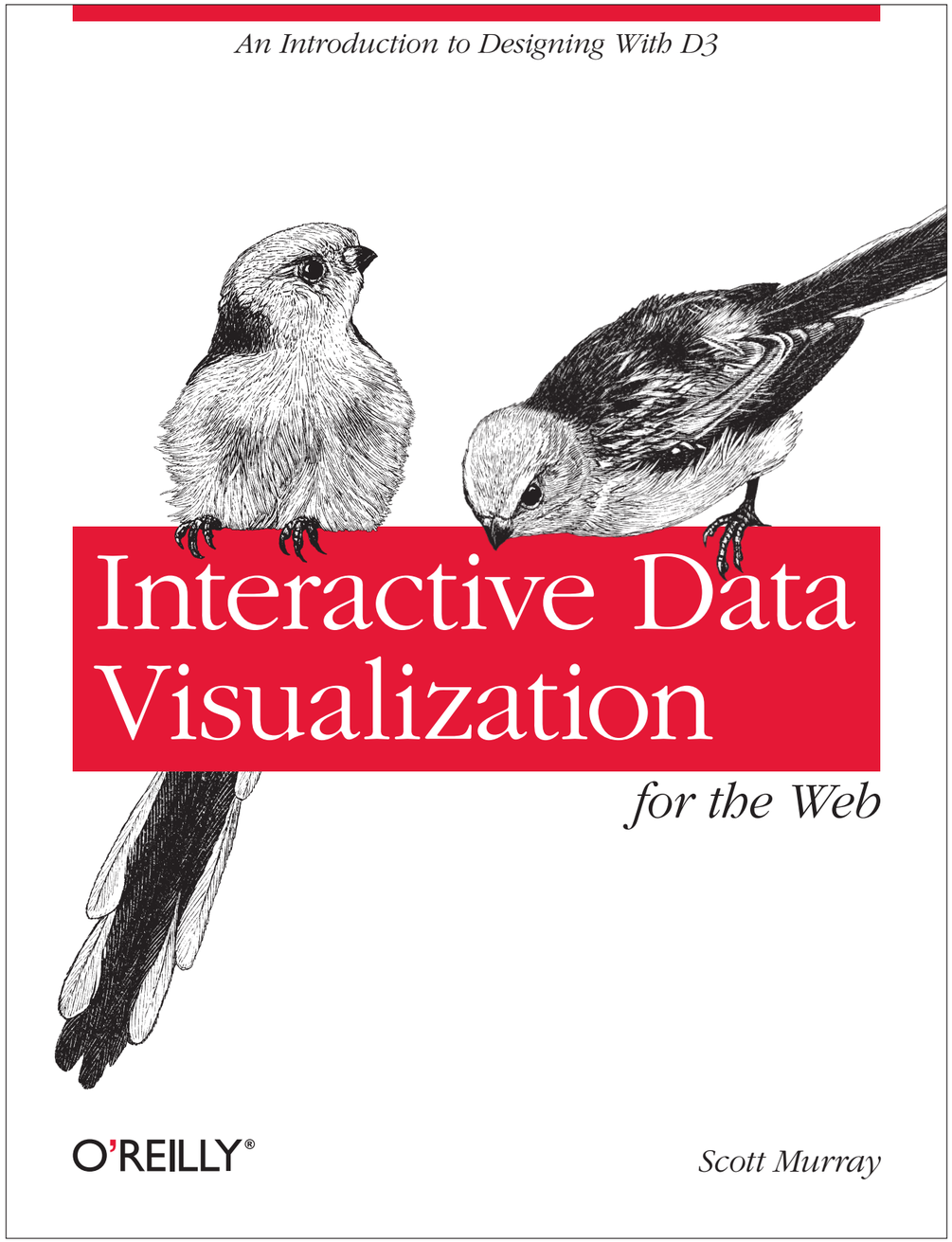


// Geomapping and projections

Jérôme Cukier

@jcukier

jeromecukier.net



Book signing today!

5:30 pm
O'Reilly Booth
Expo Hall

Scott Murray

@alignedleft

alignedleft.com