

Data Structures D3.js Accepts

The Goal

In this section, you will understand how to structure data for better use of D3.js.

We will cover where we have been, how D3.js thinks of selections, the basics of JavaScript objects and how to best structure our data.

What we have seen thus far

Every example we have done with D3.js thus far has included a data set at the top of the example like this one:

```
1 var spaceCircles = [30, 70, 110];
```

This variable **spaceCircles** is a JavaScript Array. An Array is an enumerated list of variables. This means that each element has a index number attached to it, starting with the number **0**.

In the **spaceCircles** case if we type the following into the JavaScript Console, we'll get:

```
1 var spaceCircles = [30, 70, 110];
2
3 spaceCircles[0]
4 // returns the number 30
5
6 spaceCircles[1]
7 // returns the number 70
8
9 spaceCircles[2]
10 // returns the number 110
```

Note: the // characters designate a comment in JavaScript and are not evaluated.

JavaScript Arrays can hold any type of information you want to put into them. You can have arrays of numbers, objects, strings, arrays, HTML Elements, DOM Elements, etc.

D3.js Selections are Arrays

D3.js Selections are JavaScript Arrays.

If you run a simple selection in through the JavaScript console.log in the JavaScript Console:

```
1 console.log(d3.select("body"));
```

You get an array of 1 element - the HTML "body" element:



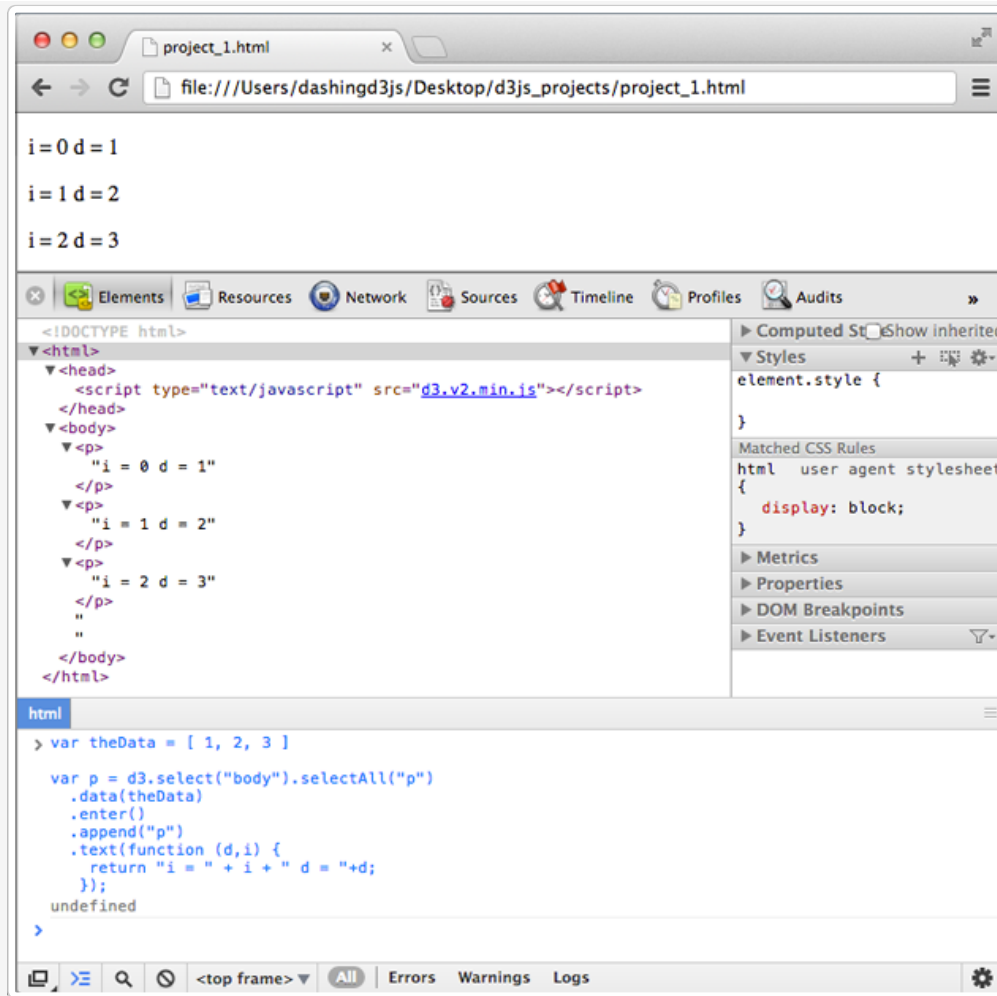
If you recall the [Using Data Bound to DOM Elements](#) section, we covered Variables Available inside D3.js Operators.

One of those variables was the **index, or i**. This variable refers to the Array index of the selection that D3.js is referencing.

Which is why the following code worked:

```
1 var theData = [ 1, 2, 3 ]
2
3 var p = d3.select("body").selectAll("p")
4   .data(theData)
5   .enter()
6   .append("p")
7   .text(function (d,i) {
8     return "i = " + i + " d = "+d;
9   });
```

Which gives us the following:



Which means that it's important to make sure that the `.data()` operator receives an array of data, regardless of what is inside of the array.

Loading External Data Resources

D3.js has built in functionality to load in the following types of external resources:

- an XMLHttpRequest
- a text file
- a JSON blob
- an HTML document fragment
- an XML document fragment
- a comma-separated values (CSV) file
- a tab-separated values (TSV) file

Each of these resources will return data that D3.js can then use. The only thing to pay attention to is to make sure you construct an array out of the data.

So far we are manually inputting our data array and will continue to do so for a while longer.

Later sections will cover how to setup a server and load the external resources.

JSON

JSON stands for JavaScript Object Notion. This is a JavaScript data structure where the indices are named.

The JSON is a collection of name/value pairs. In JSON, the name **must always be in double quotes**. As an example:

```
1 var secretAgent = {  
2   "name": "James Bond",  
3   "drink": "dry martini - shaken, not stirred",  
4   "number": "007"  
5 };
```

This allows us to get specific information from our variable:

```
1 var secretAgent = {  
2   "name": "James Bond",  
3   "drink": "dry martini - shaken, not stirred",  
4   "number": "007"  
5 };  
6  
7 secretAgent.number;  
8 // "007" is returned.
```

This makes referencing code and data much easier.

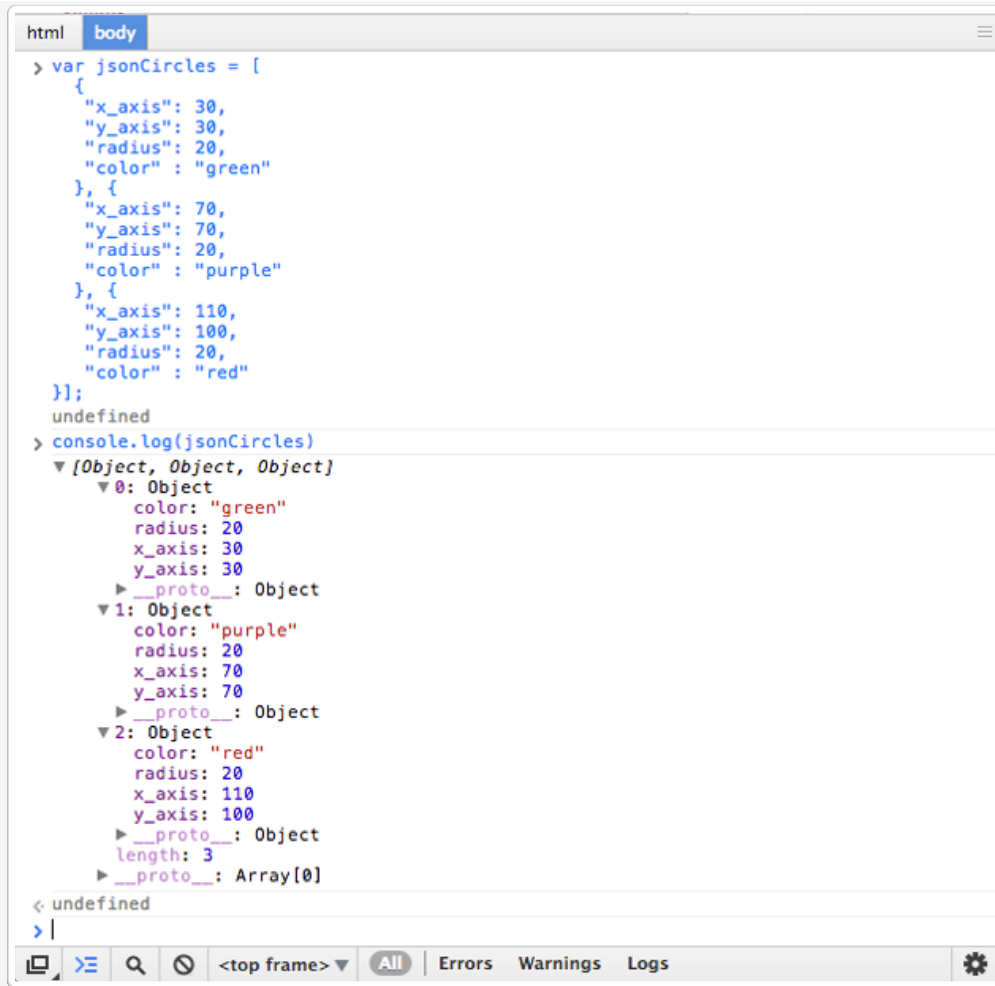
Array of JSON Objects

As we discussed at the top of this section - Arrays can hold any type of information, even JSON.

Going back to our circle examples in the previous section ([Using the SVG Coordinate Space](#)), we could write all the necessary data to visualize our circles as an array of circle objects:

```
1 var jsonCircles = [  
2   {  
3     "x_axis": 30,  
4     "y_axis": 30,  
5     "radius": 20,  
6     "color": "green"  
7   }, {  
8     "x_axis": 70,  
9     "y_axis": 70,  
10    "radius": 20,  
11    "color": "purple"  
12  }, {  
13    "x_axis": 110,  
14    "y_axis": 100,  
15    "radius": 20,  
16    "color": "red"  
17  }];
```

If we run our variable through the console.log, we can see that **jsonCircles** is an array of three objects:



```
html body
> var jsonCircles = [
  {
    "x_axis": 30,
    "y_axis": 30,
    "radius": 20,
    "color": "green"
  }, {
    "x_axis": 70,
    "y_axis": 70,
    "radius": 20,
    "color": "purple"
  }, {
    "x_axis": 110,
    "y_axis": 100,
    "radius": 20,
    "color": "red"
  }
];
undefined
> console.log(jsonCircles)
▼ [Object, Object, Object]
  0: Object
    color: "green"
    radius: 20
    x_axis: 30
    y_axis: 30
    __proto__: Object
  1: Object
    color: "purple"
    radius: 20
    x_axis: 70
    y_axis: 70
    __proto__: Object
  2: Object
    color: "red"
    radius: 20
    x_axis: 110
    y_axis: 100
    __proto__: Object
  length: 3
  __proto__: Array[0]
< undefined
> |
```

To get data out of the **jsonCircles** we can do the following:

```
1 var jsonCircles = [
2   { "x_axis": 30, "y_axis": 30, "radius": 20, "color": "green"},
3   { "x_axis": 70, "y_axis": 70, "radius": 20, "color": "purple"},
4   { "x_axis": 110, "y_axis": 100, "radius": 20, "color": "red"}];
5
6 jsonCircles[0].color;
7 // returns "green"
8
9 jsonCircles[1].y_axis;
10 // returns 70
11
12 jsonCircles[2].radius;
13 // returns 20
```

This will be extremely useful in the next and subsequent sections for use inside of functions when using D3.js operators and methods.

Want to better understand this topic? Check out this step-by-step course => [Introductory D3 Training](#)

← Using the SVG Coordinate Space

Using JSON to Simplify Code →

Learn D3.js

[D3 Tutorial](#)
[D3 Screencasts](#)
[D3 Mapping Training](#)
[D3 Introductory Training](#)
[D3 Intermediate Training](#)
[D3 Advanced Training](#)
[D3 Corporate Training](#)

DashingD3js.com

[Blog](#)
[About](#)
[Hire Me](#)
[D3 Examples](#)
[D3 Resources](#)
[D3 & Data Viz Newsletter Archive](#)

Data Visualization & D3.js Weekly Newsletter

Get D3.js and Data Visualization news, articles, jobs and more delivered to your inbox every Tuesday:

Did you sign up for the newsletter? :)

© 2012-2015 DashingD3js.com. All rights reserved.