

Brief Intro and Tutorial of D3

— — Bring data to life with SVG, Canvas and HTML. — —

Jiannan Xiao



JavaScript

- Simple grammar to learn
- IDE: WebStorm (JetBrains, education version), Debug: Chrome, Firefox
- More efficient: TypeScript, CoffeeScript
- Async and callback
 - Sync: C++, Java

A `$("p").hide("slow", function() {
 alert("hidden");
});` // alert after hide effect

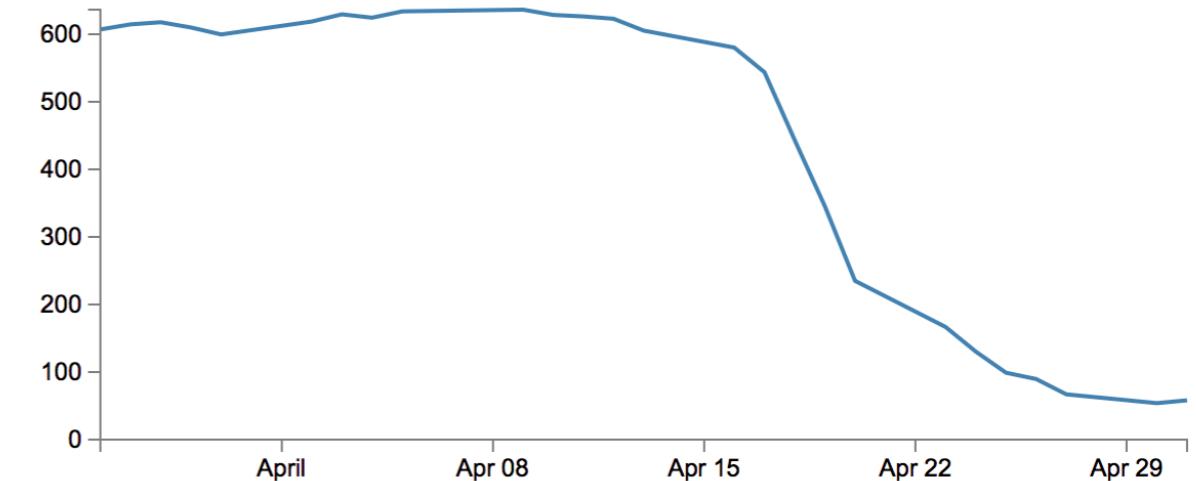
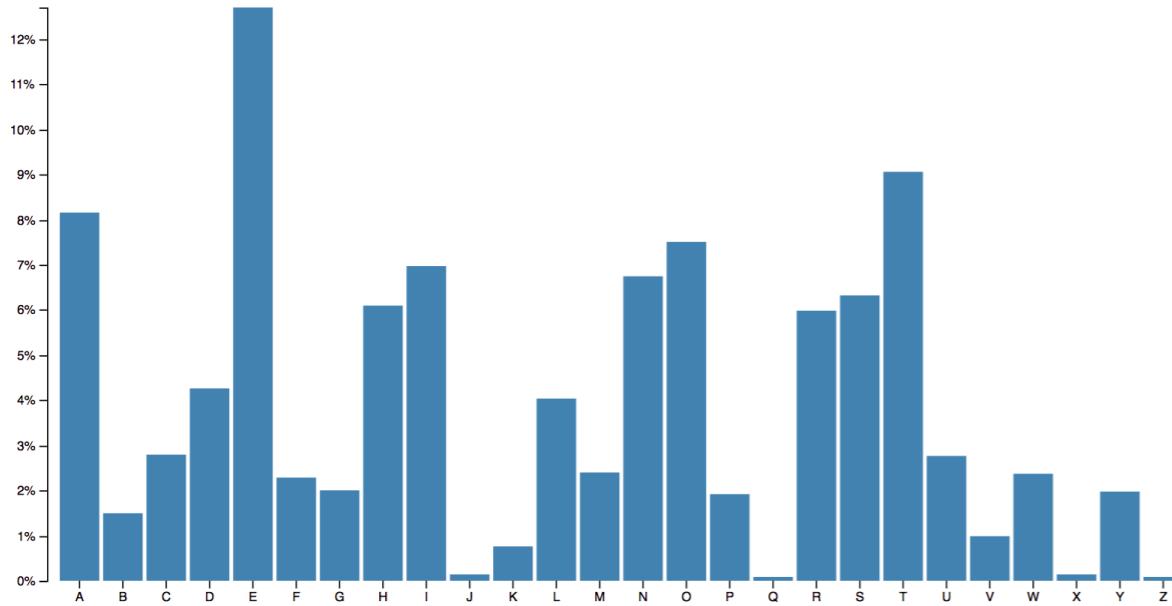
B `$("p").hide(1000);
alert("hidden");` // alert before hide effect

JavaScript Debug



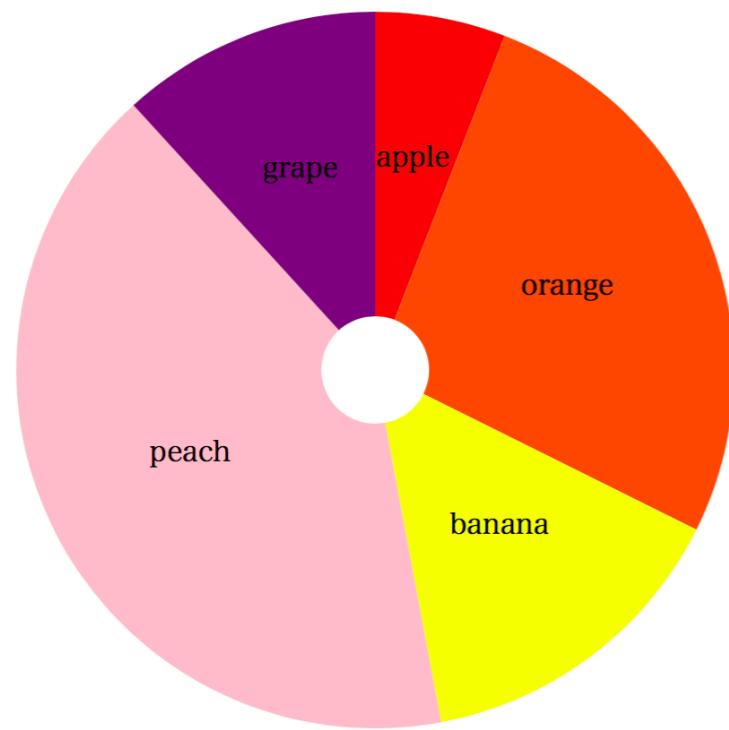
D3 Demo

- Bar chart
- <https://bl.ocks.org/mbostock/3885304>
- Line chart
- <http://bl.ocks.org/d3noob/b3ff6ae1c120eea654b5>

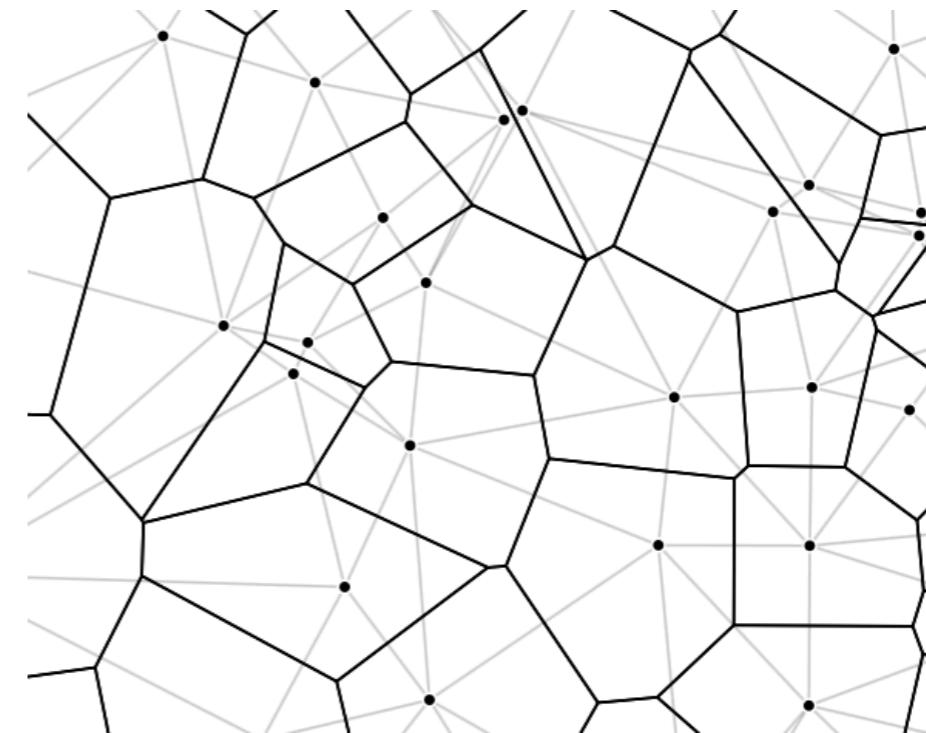


D3 Demo (cont'd)

- Pie chart
- <http://bl.ocks.org/hunzy/9134534>

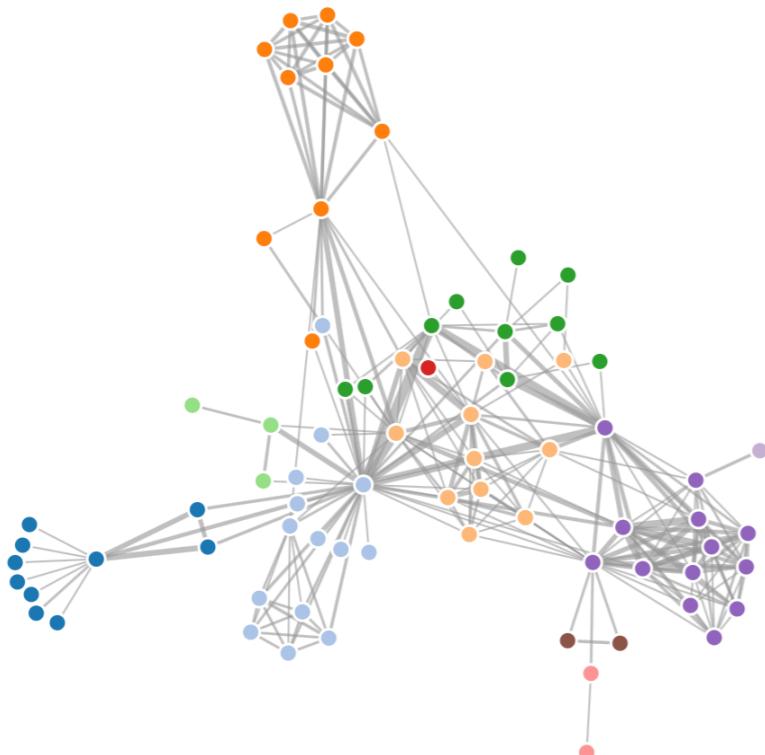


- Voronoi
- <https://bl.ocks.org/mbostock/4060366>



D3 Demo (cont'd)

- Force directed layout
- <https://bl.ocks.org/mbostock/4062045>



- Calendar view
- <https://bl.ocks.org/mbostock/4063318>

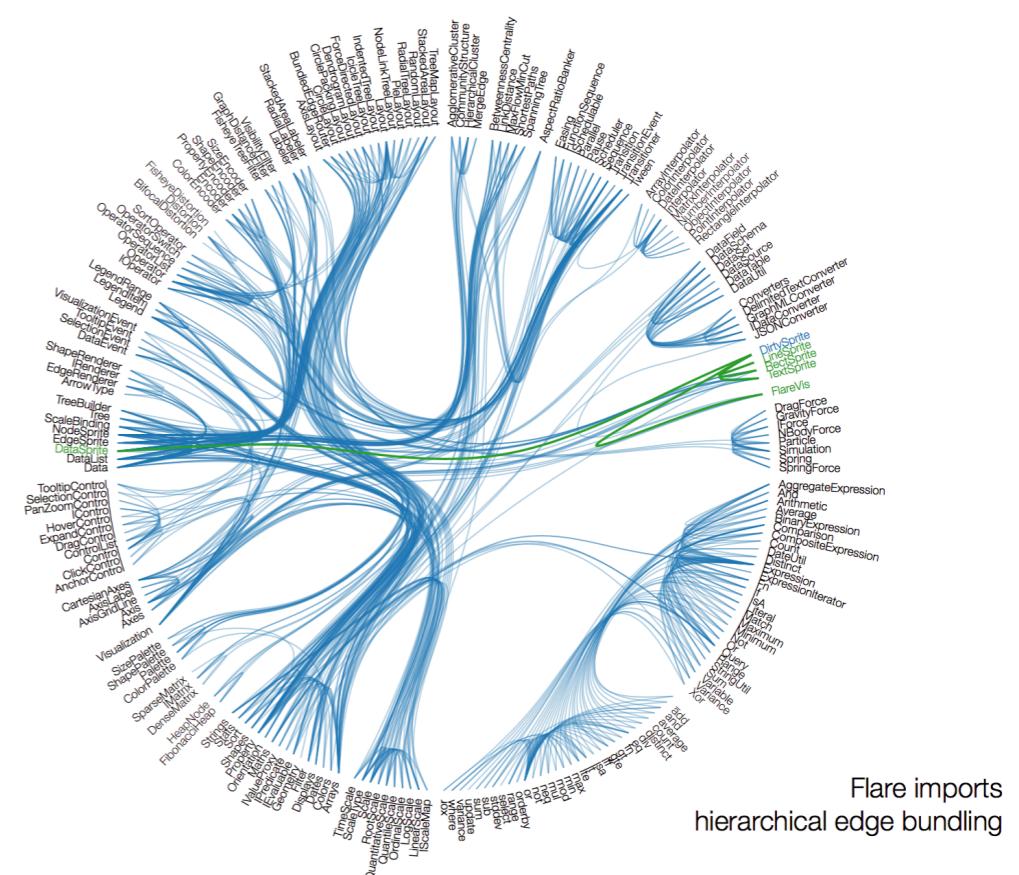


D3 Demo (cont'd)

- Sunburst
 - <https://bl.ocks.org/mbostock/4063423>



- Edge bundling
 - <http://mbostock.github.io/d3/talk/20111116/bundle.html>



More Demos

- <https://github.com/d3/d3/wiki/Gallery>
- <https://bl.ocks.org/mbostock>
- <http://christopheviau.com/d3list/gallery.html>

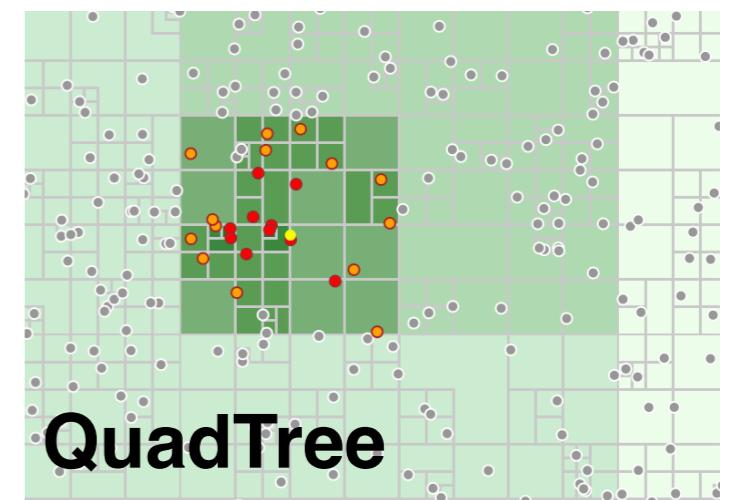
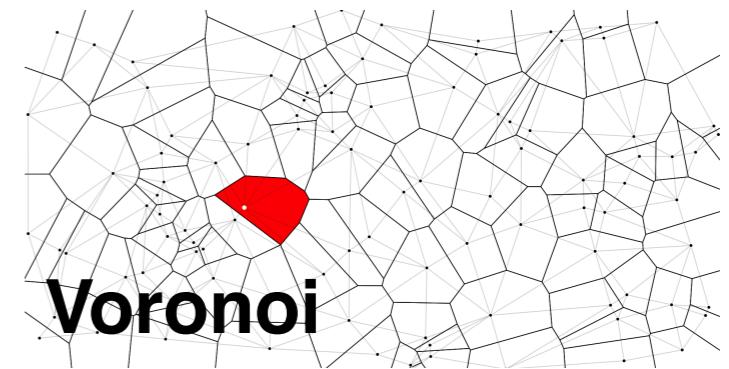
D3

- Data Driven Documents
 - Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. "D³ data-driven documents." IEEE transactions on visualization and computer graphics 17.12 (2011): 2301-2309.
- A JavaScript library binding arbitrary data to DOM (Document Object Model)
- Based on SVG, instead of Canvas
- Inspired libraries
 - NVD3, C3.js, N3

```
▼ <svg width="960" height="960"
"10" text-anchor="middle">
  ▼ <g class="node" transform=
"translate(445.33474420445697,
<circle id="flare.analytics
"24.996944488871147" style=
  ▶ <clipPath id="clip-flare.ar
...</clipPath>
  ▶ <text clip-path="url(#clip-
flare.analytics.cluster.Aggl
<title>flare.analytics.clus
3,938</title>
</g>
  ▶ <g class="node" transform=
"translate(496.4261103521095,4
  ▶ <g class="node" transform=
"translate(471.3454728058744,5
```

Modules

- SVG
 - Line, Arc, Area, Chord
- Layouts
 - Chord, Bundle, Force, Tree, Treemap
- Geometry
 - Voronoi, QuadTree
- Scale, time, behaviors



D3 Hello World!

```
<!DOCTYPE html>
<meta charset="utf-8">
<body>
<script src="//d3js.org/d3.v3.min.js"></script>
<script>

d3.select("body").append("span")
    .text("Hello, world!");

</script>
```

Hello, world!

Selection

- CSS selector like grammar (partial support)
 - `#some-id`
 - `.some-class`
 - `some-tag-name`
- `d3.selectAll`: return all that matches
- `d3.select`: return the first that matches

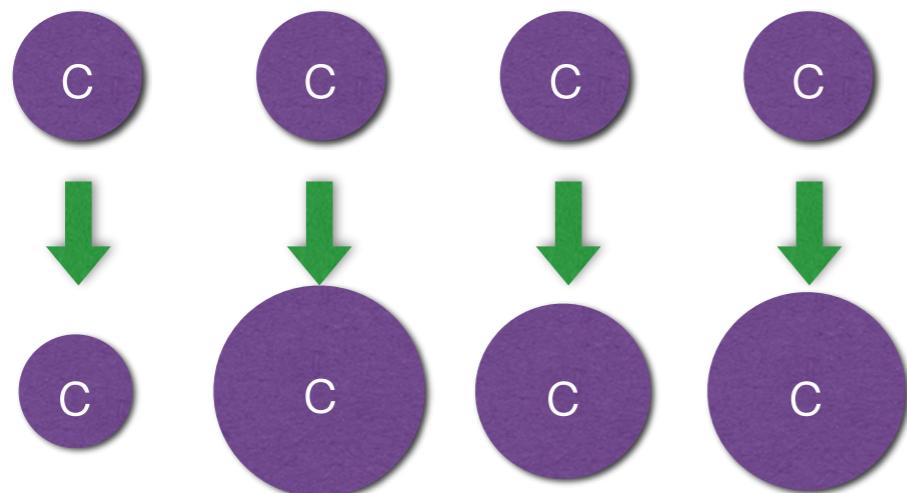
Dynamic Properties (CSS)

```
d3.selectAll("p").style("color", function() {
  return "hsl(" + Math.random() * 360 + ",100%,50%)";
});
```



Dynamic Properties (SVG)

```
d3.selectAll("circle").attr("r", function(d) {  
    return Math.random() * 10;  
});
```



- 1 d3.selectAll("circle")
- 2 .attr("r", function(d) {
 return Math.random() * 10;
});

Property Source

- A**

```
d3.selectAll("circle").attr("r", function(d) {
    return Math.random() * 10;}) ;
```

 <circle r=10></circle>

- B**

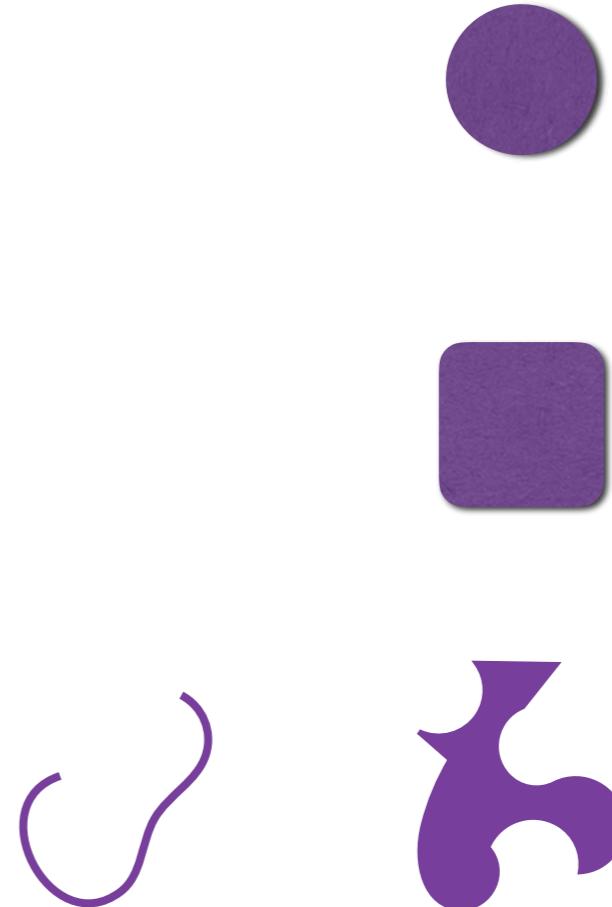
```
d3.selectAll("circle").style("r", function(d) {
    return Math.random() * 10;}) ;
```

 <circle style="r: 10"></circle>

- CSS properties and SVG properties
 - CSS: background-color, font-size
 - SVG: a little complex

SVG Properties

- Look up in [MDN web docs](#)
- Common: stroke, fill, stroke-width
- Circle
 - cx, cy, r
- Rect
 - x, y, width, height, rx, ry
- Path (open, closed)
 - d
 - <https://www.jasondavies.com/animated-bezier/>



Enter and Exit

radius_list [4, 8, 15, 16, 23, 42] → a list of circles

1 d3.selectAll("circle") bind DOM elements

2 d3.selectAll("circle").data([4, 8, 15, 16, 23, 42]) bind data

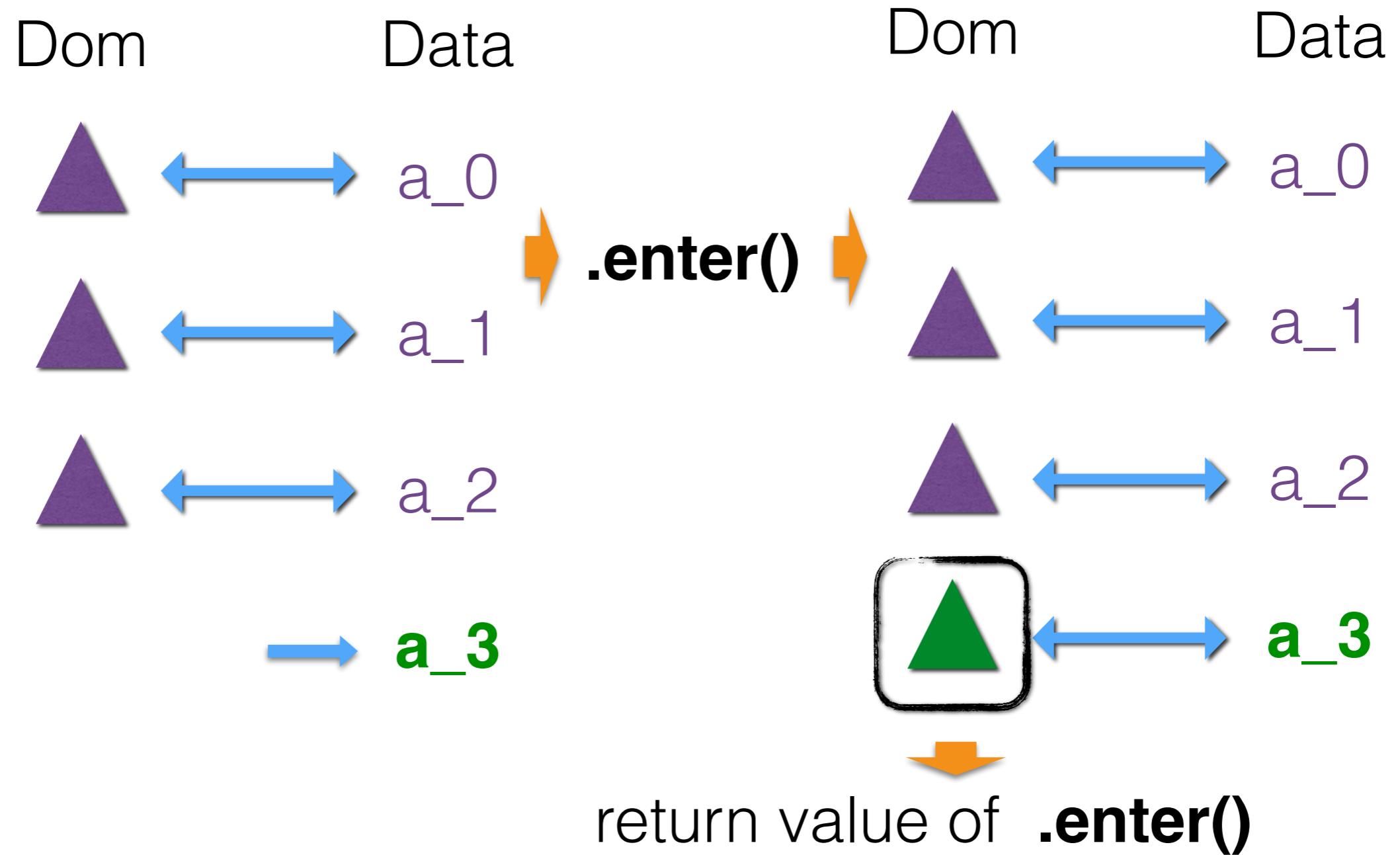
3 d3.selectAll("circle").data([4, 8, 15, 16, 23, 42]).enter().append("circle")

create new nodes
for incoming data

4 d3.selectAll("circle").data([4, 8, 15, 16, 23, 42]).enter().append("circle").attr("r", function(d) {
 return d;
});

set radius

Enter and Exit (cont'd)



Tutorials

- Code
 - <https://github.com/Derek-Xiao/d3-demos>
- Tree, Scatterplot, Voronoi

Tutorial: Tree

- Step 1: Create container

```
var chart = d3.select("body").append("svg")
  .attr("width", 960)
  .attr("height", 960)
  .append("g")
  .attr("transform", "translate(50, 400)");
```

- Effect

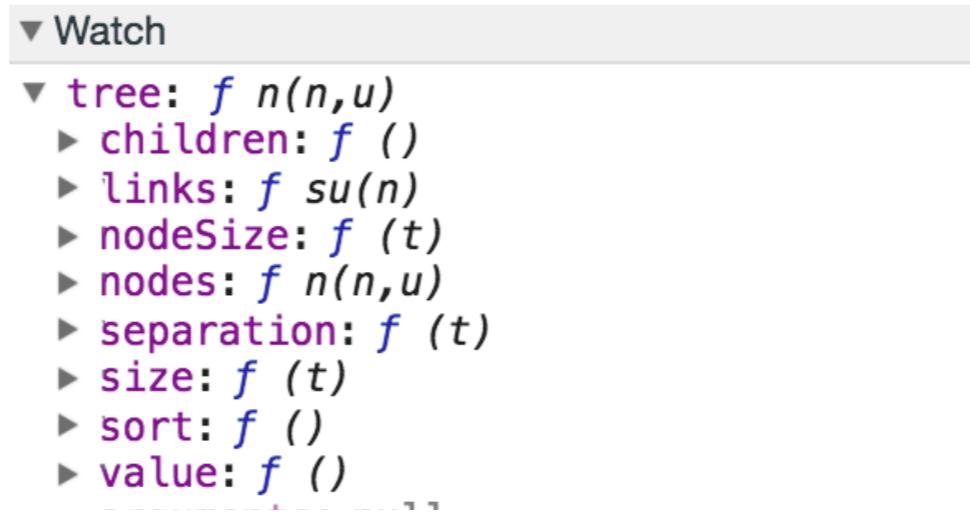
```
▼ <svg width="960" height="960">
  <g transform="translate(50, 400)"></g>
</svg>
```

Tutorial: Tree

- Step 2: Create tree layout

```
var tree = d3.layout.tree()  
    .size([860, 400])  
    .nodeSize([20, 120]);
```

- Effect



Tutorial: Tree

- Step 3: Create tree layout

```
d3.json("../data/tree.json", function(data) {  
  var nodes = tree.nodes(data);  
  var links = tree.links(nodes);  
});
```

- Effect

```
▼ nodes: Array(51)  
  ▶ 0: {name: "flare", children: A  
  ▶ 1: {name: "util", children: Ar  
  ▶ 2: {name: "Arrays", size: 8258  
  ▼ 3:  
    depth: 2  
    name: "Colors"  
    ▶ parent: {name: "util", childr  
    size: 10001  
    x: -332.5  
    y: 240  
  ▾ links: Array(50)  
    ▾ 0:  
      ▶ source: {name: "flare", child  
      ▶ target: {name: "util", childr  
      ▶ __proto__: Object  
      ▶ 1: {source: {...}, target: {...}}  
      ▶ 2: {source: {...}, target: {...}}  
      ▶ 3: {source: {...}, target: {...}}  
      ▶ 4: {source: {...}, target: {...}}  
      ▶ 5: {source: {...}, target: {...}}  
      ▾ 6: {source: {...}, target: {...}}
```

Tutorial: Tree

- Step 4: Create tree node holder

```
var nodes = chart.selectAll(".node")
  .data(nodes).enter()
  .append("g")
  .attr("class", "node")
  .attr("transform", function(d){
    return "translate(" + d.y + "," + d.x +
  ")");
}

nodes.append("circle")
  .attr("r", 5)
  .attr("fill", "steelblue");
```

- Effect

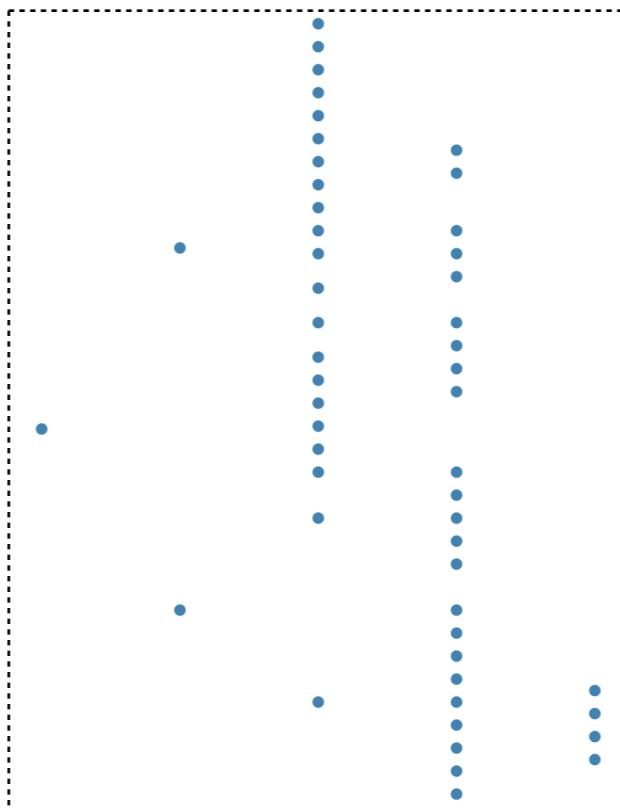
```
▼ <svg width="960" height="960">
  ▼ <g transform="translate(50, 400)">
    <g class="node" transform="translate(0,0)"></g>
    <g class="node" transform="translate(120,-157.5)"></g>
    <g class="node" transform="translate(240,-352.5)"></g>
    <g class="node" transform="translate(240,-332.5)"></g>
```

Tutorial: Tree

- Step 5: Create tree nodes

```
nodes.append("circle")
    .attr("r", 5)
    .attr("fill", "steelblue");
```

- Effect



Tutorial: Tree

- Step 6: Create tree node text

```
nodes.append("text")
  .text(function(d){ return d.name; })
  .attr("text-anchor", "start")
  .attr("x", function(d) {
    return d.children || d._children ? -13 : 13;
  })
  .attr("text-anchor", function(d) {
    return d.children || d._children ? "end" : "start";
  })
  .attr("dy", ".35em");
```

- Effect

```
▼ <svg width="960" height="960">
  ▼ <g transform="translate(50, 400)">
    ▼ <g class="node" transform="translate(0,0)">
      <circle r="5" fill="steelblue"></circle>
      <text text-anchor="end" x="-13" dy=".35em">flare</text>
    </g>
```

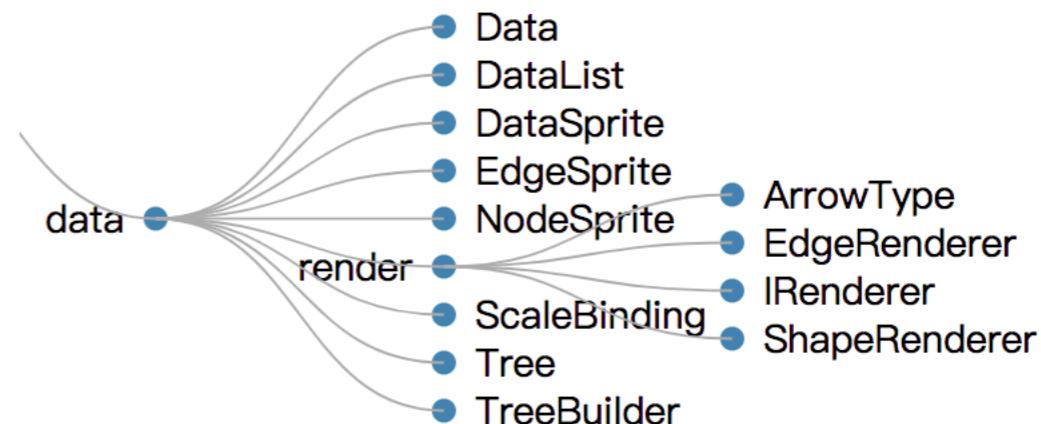
- Arrays
- Colors
- Dates
- Displays
- Filter
- Geometry
- heap
- IEvaluable
- IPredicate
- IValueProxy
- math
- Maths
- Orientation

Tutorial: Tree

- Step 7: Create tree edges

```
var diagonal = d3.svg.diagonal()  
    .projection(function(d){ return [d.y, d.x]; })); // flip x and y of links  
  
chart.selectAll(".link")  
    .data(links).enter()  
    .append("path")  
    .attr("class", "link")  
    .attr("fill", "none").attr("stroke", "#ADADAD")  
    .attr("d", diagonal);
```

- Effect



```
<path class="link" fill="none" stroke="#ADADAD" d="M0,0C60,0 60,-157.5 120,-157.5"></path>  
<path class="link" fill="none" stroke="#ADADAD" d="M0,0C60,0 60,157.5 120,157.5"></path>  
<path class="link" fill="none" stroke="#ADADAD" d="M120,-157.5C180,-157.5 180,-352.5 240,-352.5"></path>  
<path class="link" fill="none" stroke="#ADADAD" d="M120,-157.5C180,-157.5 180,-332.5 240,-332.5"></path>
```

Resources

- D3 home project: <https://github.com/d3/d3>
- Tutorials
 - <https://www.dashingd3js.com/table-of-contents>
 - http://www.gtz.com/docs_books/Getting_Started_with_D3.pdf
 - <http://techslides.com/over-1000-d3-js-examples-and-demos>
- Rich demos and examples
 - <https://github.com/d3/d3/wiki/Gallery>
 - <https://bl.ocks.org/mbostock>
 - <http://christopheviau.com/d3list/gallery.html>

Snack for U

- How to simulate Conway's Game of Life using D3
 - https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

