Web Mapping and Data Visualization with Leaflet and D3

Shruti Mukhtyar

Spatial Data Science Bootcamp

@mapchitra @cal_adapt

Where to Begin?

Quick Review:

- Tools - Code Editor & Chrome Developer Tools
- HTML
- CSS
- DOM
- JavaScript

Leaflet:

- What is Leaflet?
- Exercises
- Review

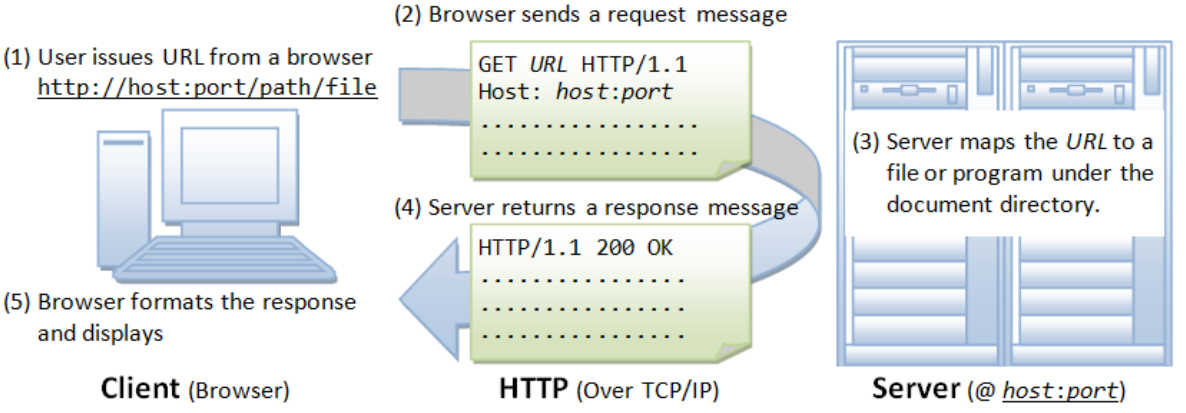
D3:

- What is D3?
- SVG, Canvas
- Explore

Tools for Web Development

- Code Editor
  - Sublime Text, Atom, PyCharm, Vim, Notepad++, IDLE, Gedit, TextMate
  - Syntax highlighting, indentation, autocomplete, bracket matching
  - Run interpreters, debuggers
- Browser
  - Developer Tools - press `Ctrl+Shift+I` (or `F12`) in Chrome
  - More info on Chrome Developer tools here

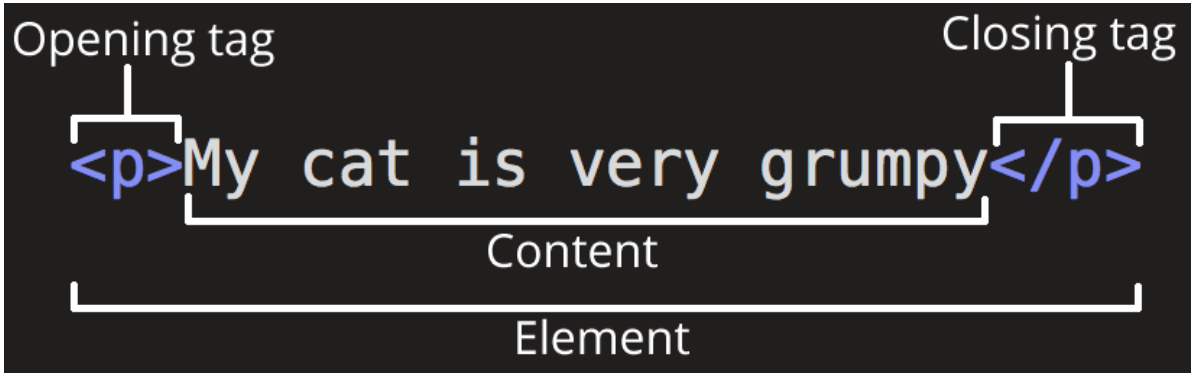
Quick Review of Client & Server



- HTTP, language of the web
- Browser sends HTTP GET Request. Server sends response for each request with content and/or status message.
- HTTP Basics

Hyper Text Markup Language (HTML)

- HTML is not a programming language; it is a markup language
- Role » Describes content
- HTML Elements consist of a opening tag, the content, and a closing tag




```
<ul>
  <li><a href="gif.berkeley.edu">Home Page</a></li>
  <li><a href="gif.berkeley.edu/people">Staff</a></li>
</ul>
<div id="map" class="center" style="height:500px;"></div>
```

- HTML5 Resources
  - Dive into HTML5
  - Mozilla Developer Network

Cascading Style Sheet (CSS)

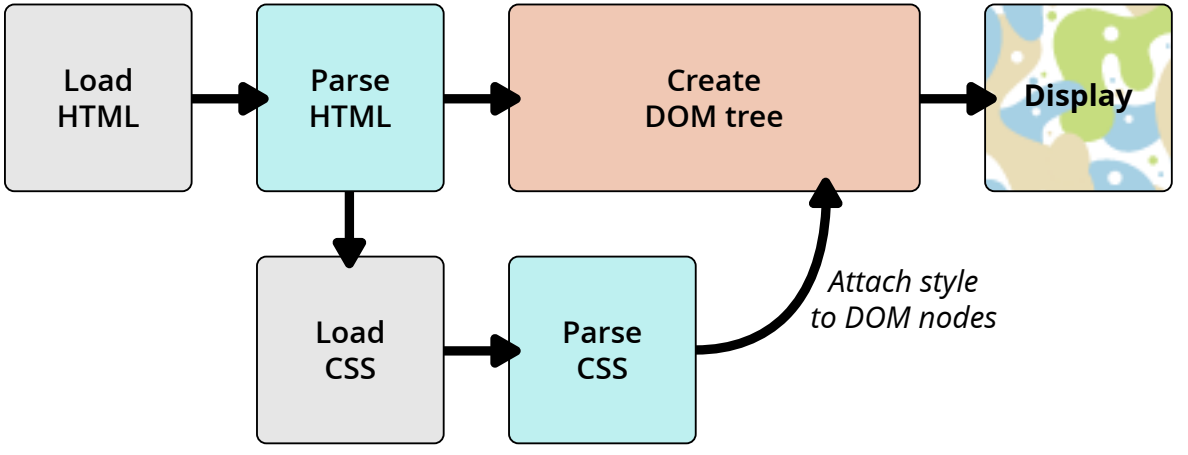
- Not a programming or markup language
- Role » Describes presentation of a document written in HTML or XML (e.g. SVG)
- Selectors, properties, and values
- Properties and Attributes

```
#map {
  height: 500px;
}
h1 {
  font-family: "Helvetica", "sans-serif";
  font-weight: bold;
  background-color: #000;
}
.chart {
  float: left;
}
```

- CSS3 Resources
  - Mozilla Developer Network

Document Object Model (DOM)

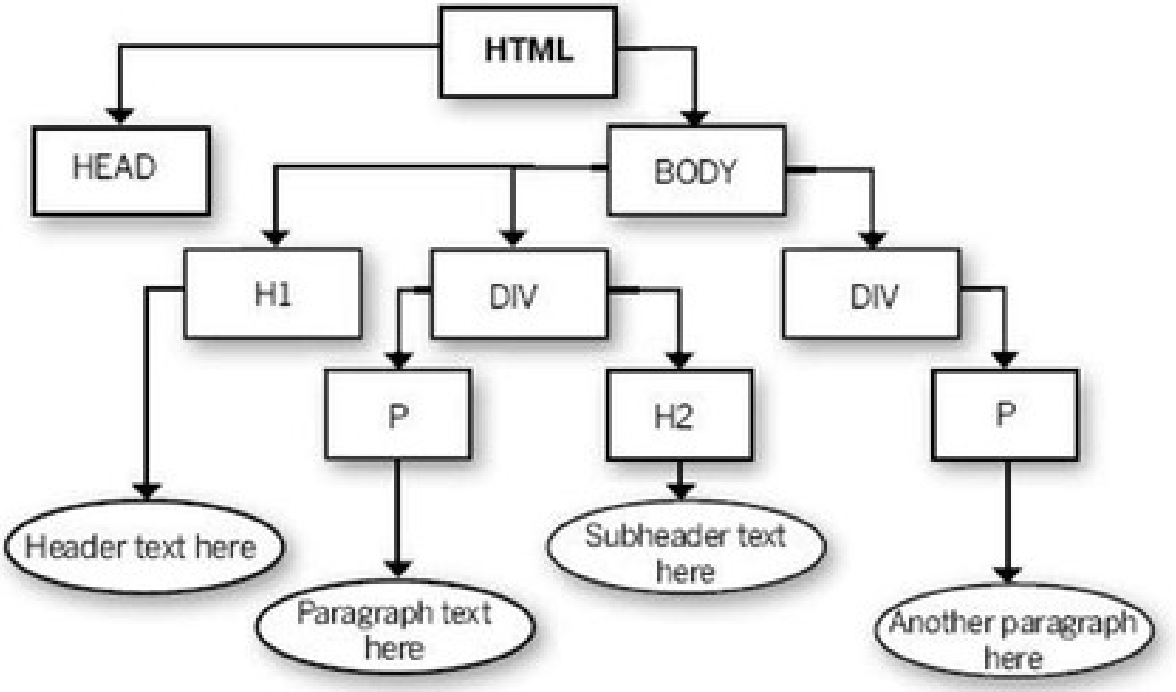
- Structured representation of the document (a tree) created by the browser
- HTML you write is parsed by the browser and turned into the DOM
- Programming interface for HTML and SVG documents
- JavaScript manipulates the DOM



Conceptual Web Page

```
<!DOCTYPE html>
<html>
<head>
    <!-- head content -->
    <style type="text/css">
      div {
        color: red;
      }
    </style>
</head>
<body>
    <h1>Header text here</h1>
    <div>
        <h2>Subheader text here</h2>
        <p>Paragraph text here</p>
    </div>
    <div>
        <p>Another paragraph here</p>
    </div>
    <script>
        <!-- JavaScript content -->
    </script>
</body>
</html>
```

The DOM for conceptual web page



JavaScript

- "Easy to learn, hard to master"
- Role » Creating interaction
- Interpreted by your browser
- Asychronous (code executes in background after client-browser recieves data from server)
- Get familiar with
  - Working with json objects
  - Array functions (forEach, filter, map)
  - Method chaining, Callbacks, Closures, Modules
- JS Resources
  - Mozilla Developer Network

Codepen.io

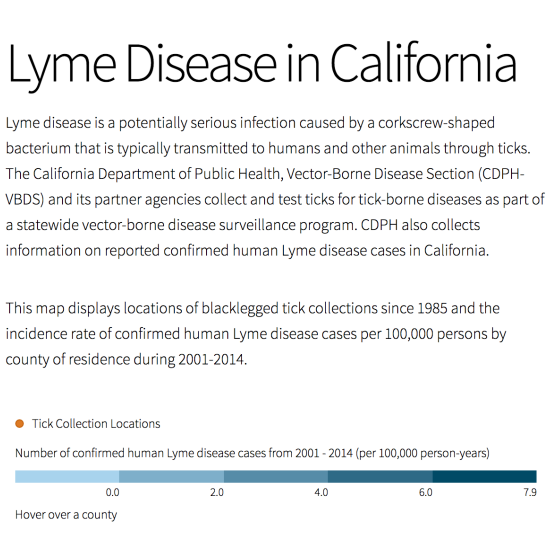
- Playground for the front-end web developmet
- Open this link in a new tab to explore how HTML, CSS and JavaScript work together.
- Some other playgrounds
  - JSFiddle
  - JS Bin
  - Blockbuilder - D3

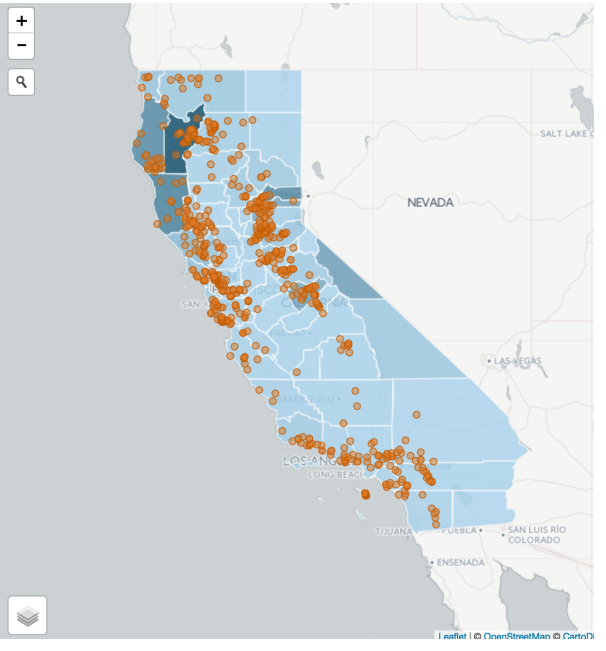
JavaScript Resources

- If you are going to develop medium to large scale web apps learn more about using front-end development tools.
  - Getting Started Web Development Guide
  - Front-end Handbook
- Google Search
  - Favor results from Stack Exchange, Mozilla Developer Network, CSS-Tricks
- caniuse - Check which browsers support what features
- Web Design
  - A List Apart
  - Webdesigner News - curated stories
- Eloquent Javascript
- Egghead.io - Bite size web dev video training
- Lynda.com, Frontend Masters, CodeAcademy, Udacity, Coursera, etc.

Preview of what we'll be building

- Click on `final` directory and open in new tab





Leaflet

Leaflet

- Lightweight, simple & flexible open source JavaScript mapping library
- Created by Vladimir Agafonkin. Great speaker, checkout some of his talks on Leaflet
- Mobile-friendly. Well documented API, huge amount of plugins.
- Use with other JS mapping libraries (like Esri-Leaflet) or by itself. Similar libraries - OpenLayers, ModestMaps, Polymaps.
- Carto.js and Mapbox.js libraries are built on top of Leaflet.
- Leaflet FAQ

What does it do?

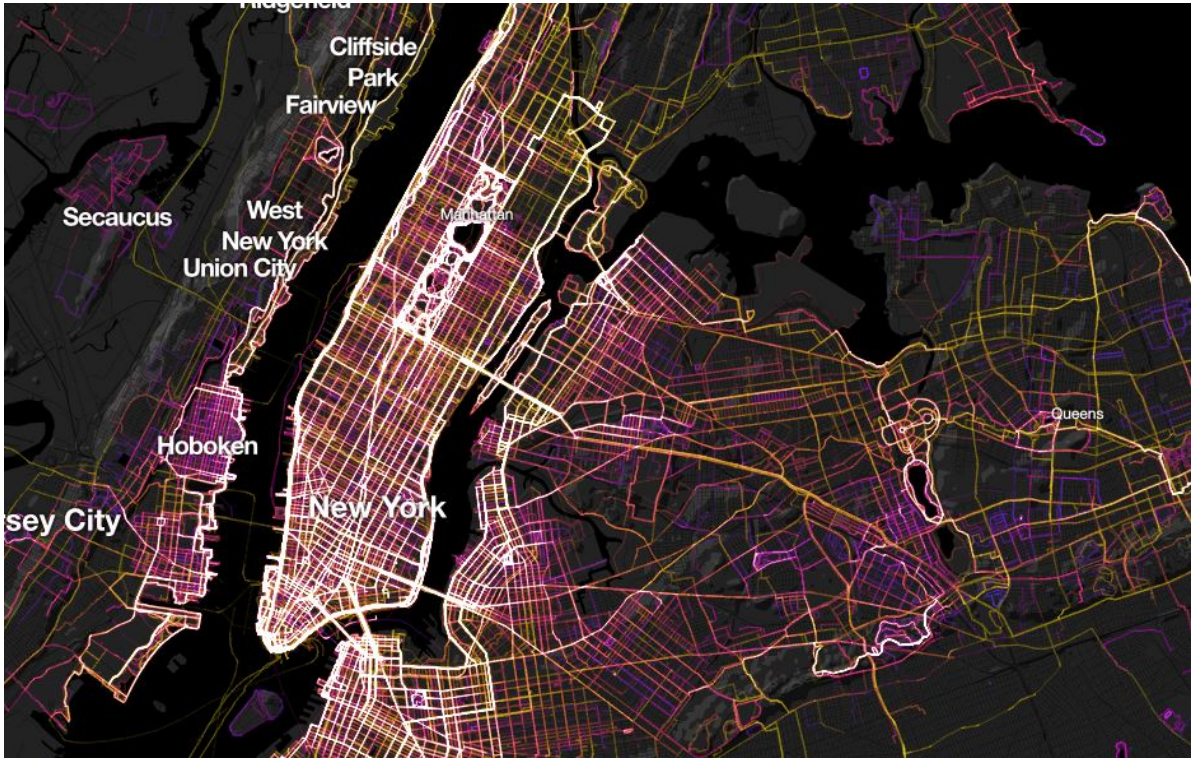
- Slippy maps with panning and zooming
- Provides functions for converting data into map layers
- Provides mouse interaction
- Does not provide any data
- You provide tile basemaps and data for overlays
- Easy to extend with plugins

Overlays - GeoJSON

```
var obj =
{ "type": "FeatureCollection",
    "features": [
      { "type": "Feature",
        "geometry": {"type": "Point", "coordinates": [102.0, 0.5] },
        "properties": {"prop0": "value0"}
       },
      { "type": "Feature",
         "geometry": {
           "type": "Polygon",
           "coordinates": [
             [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],
               [100.0, 1.0], [100.0, 0.0] ]
             ]
         },
         "properties": {
           "prop0": "value0",
           "prop1": {"this": "that"}
          }
      }
    ]
}
```

- GeoJSON Specification, Validate your geojson
- Convert from shapefile - geojson.io, Mapshaper, GIS
- GeoJSON data coordinates should be in latitude, longitude (WGS 84)!!!

Basemaps & Overlays - Map Tiles



- How Web Maps Work
- Web Mercator - Standard projection for map tiles

Leaflet Exercises

- Link to Github Repo
- Before we begin:
  - Start a local server
  - Code Editor and Browser
  - Workflow for the exercises

Exercise 01 - Make a map

- This exercise covers the basics for creating a Leaflet map element and adding a basemap to it.
- Four things you absolutely need for adding Leaflet map to a web page
  - Leaflet CSS file
  - Leaflet JavaScript file
  - A div element with an `id` attribute
  - A `css` height for your map div

Exercise 02 - Add Overlay

- Adding an overlay of tick collection point locations

Exercise 03 - Style Overlay

- Experiment with applying custom styles to geojson overlay
- Leaflet allows you to pass a variety of options to `L.geoJson` to style your layer. The options are written as callback functions.
- An example of a callback function

```
// Callback Example

function mySandwich(param1, param2, callback) {
  alert('Started eating my sandwich.\n\nIt has: ' + param1 + ', ' + param2);
  callback();
}

mySandwich('ham', 'cheese', function() {
  alert('Finished eating my sandwich.');
});
```

Exercise 04 - Working with Larger Datasets

- Geojson files with lots of data can take longer to load depending on size and network speed. And on the web you never want to keep your user waiting :-) There are couple of ways you can deal with larger files.
- Option 1 - Use the topojson format. Topojson is an extension of GeoJSON that encodes topology.
    - Convert geojson to topojson using mapshaper.org. You can also further reduce file size by simplifying the geometry. It depends on your app, how much detail you need to show.
- Option 2 - Host the data on service providers like CARTO, ArcGIS Online, Mapbox. These companies also provide their own web mapping libraries (some built on top of Leaflet)
- Great mapmakers cheatsheet on other options for dealing with large files with Leaflet and web maps in general.

Exercise 05 - Add a Leaflet plugin for Geocoding

- Huge list of Leaflet plugins to extend functionality.
- We will add a geocoding plugin called Leaflet Control Geocoder

Exercise 06 - Spatial Analysis in your Browser

- Experiment with doing some spatial analysis directly in your browser using Turf.js
- Turf.js is a javascript library for spatial analysis. MapBox has a great intro on using Turf that you can check out here.

Leaflet Review

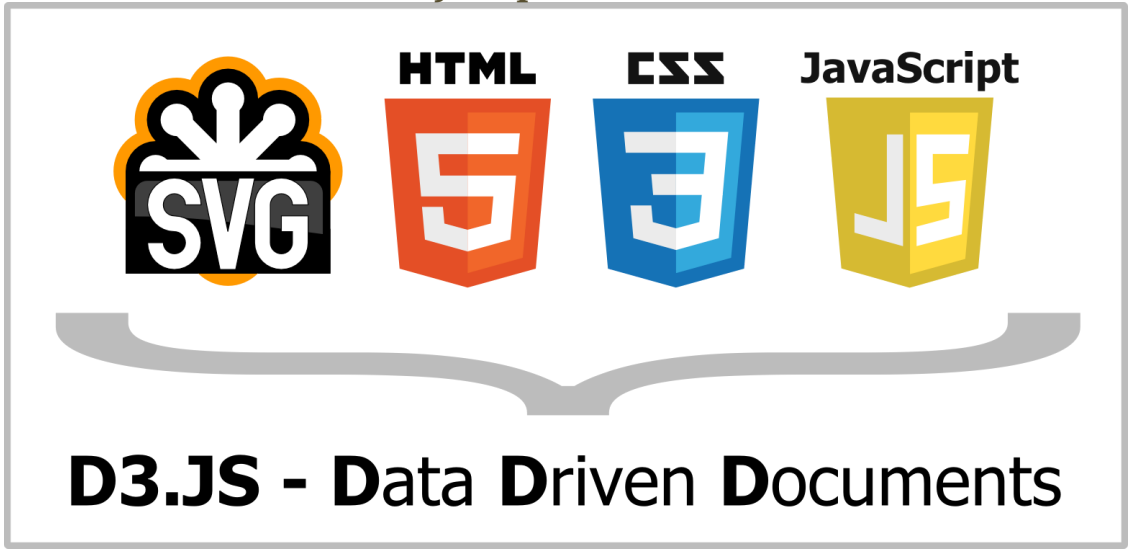
- Adding content and styling to make it look like a real website
- Adding legends
- An example of using the MarkerCluster Leaflet plugin
- An example of using the Turf.js buffer functionality

D3

Data-Driven Documents

D3 (Data Driven Documents)

- D3.js is a javascript for producing dynamic, interactive data visualizations in web browsers.
- Makes use of the widely implemented SVG, HTML5, and CSS3 standards



- Created by Mike Bostock, Vadim Ogievetsky and Jeffrey Heer (all at that time were part of the Stanford Visualization Group).

Learning D3

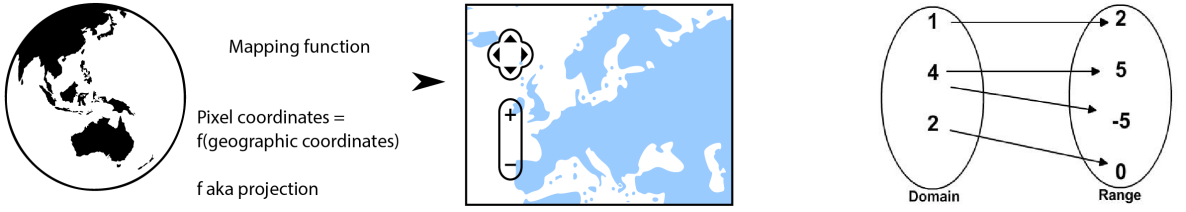
- Very active user community in Bay Area. Checkout Bay Area d3 User Group.
- Rich ecosystem of examples showing visualization types, coding techniques:
  - blockbuilder.org/
  - bl.ocks.org/
- Navigating the D3 Ecosystem
- How do I learn D3.js? Quora thread

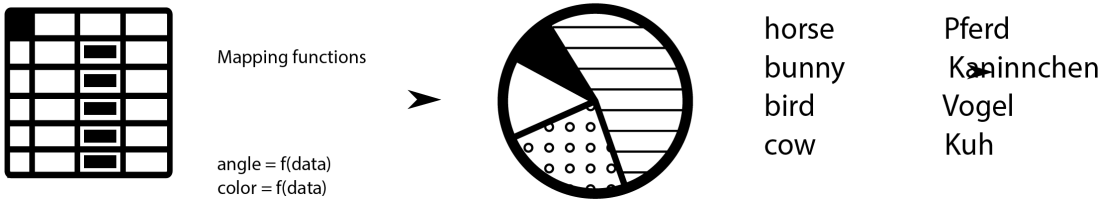
What does it do?

- Transforms data into information
- It is *not* a graphics library, it is *not* a charting library, it is a general purpose data visualization library
- Visualizes data using web standards (HTML, Javascript, CSS, DOM, SVG)
- Heavily used in data journalism and visualization. New York Times has been the pioneer in data visualization and data journalism.
- Provides new ways to think about map mapking, communication.

Revisiting mapping

- Mapping: An operation that associates each element of a given set (the domain) with one or more elements of a second set (range)





- Any (computer) visualization problem can be described as a mapping problem. Geospatial mapping is just a subset.

SVG

```
<svg width="300" height="180">
  <rect x="10" y="20" width="20" height="50" fill="blue"
              stroke="red" stroke-width="1"/>

  <circle cx="100" cy="100" r="25" fill="red"
              stroke="#ddd" stroke-width="5"></circle>

  <g transform="translate(5, 15)">
    <text x="0" y="0">My graphic</text>
  </g>

  <g transform="translate(5, 55)">
    <!-- M: move to (jump)
         L: line to
         Q: curve to (quadratic) -->
    <path d="M0,50 L50,0 Q100,0 100,50"
      fill="none" stroke-width="3" stroke="black" />
  </g>

  <g transform="translate(5, 105)">
    <!-- C: curve to (cubic)
         Z: close shape -->
    <path d="M0,100 C0,0 25,0 125,100 Z" fill="black" />
  </g>
</svg>
```

- Play with this code on SVG Graphic Primitives - JSFiddle

SVG v/s Canvas

- HTML5 Canvas is a raster based format for drawing on the web
  - You can draw raster graphcis with D3
  - You can only get pixel values on click event
  - Faster
- SVG (Scalable Vector Graphics) is a vector based format for drawing on the web
  - HTML has div and span, etc.; SVG has circle and rect, etc.
  - SVG is a DOM for graphical elements
  - You can attach events to SVG elements
  - Most D3 examples work with SVG

Dataviz Resources

- Curated lists of data viz and data viz research
  - visualisingdata.com
- Mike Bostock's blog
- Narrative Patterns for Data-Driven Storytelling
- Geographic visualizations with D3
  - An Ode to Projections
  - Jason Davies, Jason Davies Code Blocks
- **d3.v3.js ≠ d3.v4.js**

D3 - Main Concepts

- Uses pre-built JavaScript functions to select elements, create SVG objects, style them, or add transitions (dynamic effects) or tooltips
- Objects are styled with CSS
- Bind data to SVG elements by using simple D3 functions
- Most common data formats - csv, geoJSON

D3 - Main Concepts

- SELECTIONS - D3 provides shorthand for selecting and manipulating DOM objects.

```
d3.selectAll("p")                 // select all <p> elements
.style("color", "lavender")     // set style "color" to value "lavender"
.attr("class", "squares")       // set attribute "class" to value "squares"
.attr("x", 50);                 // set attribute "x" (horizontal position) to value 50px
```

- TRANSITIONS - Smoothly interpolate attributes and styles over a certain time

```
d3.selectAll("p")               // select all <p> elements
.transition("trans_1")        // transition with name "trans_1"
  .delay(0)                   // transition starting 0ms after trigger
  .duration(500)              // transitioning during 500ms
  .ease("linear")             // transition easing progression is linear...
.style("color", "pink");      // ... to color:pink
```

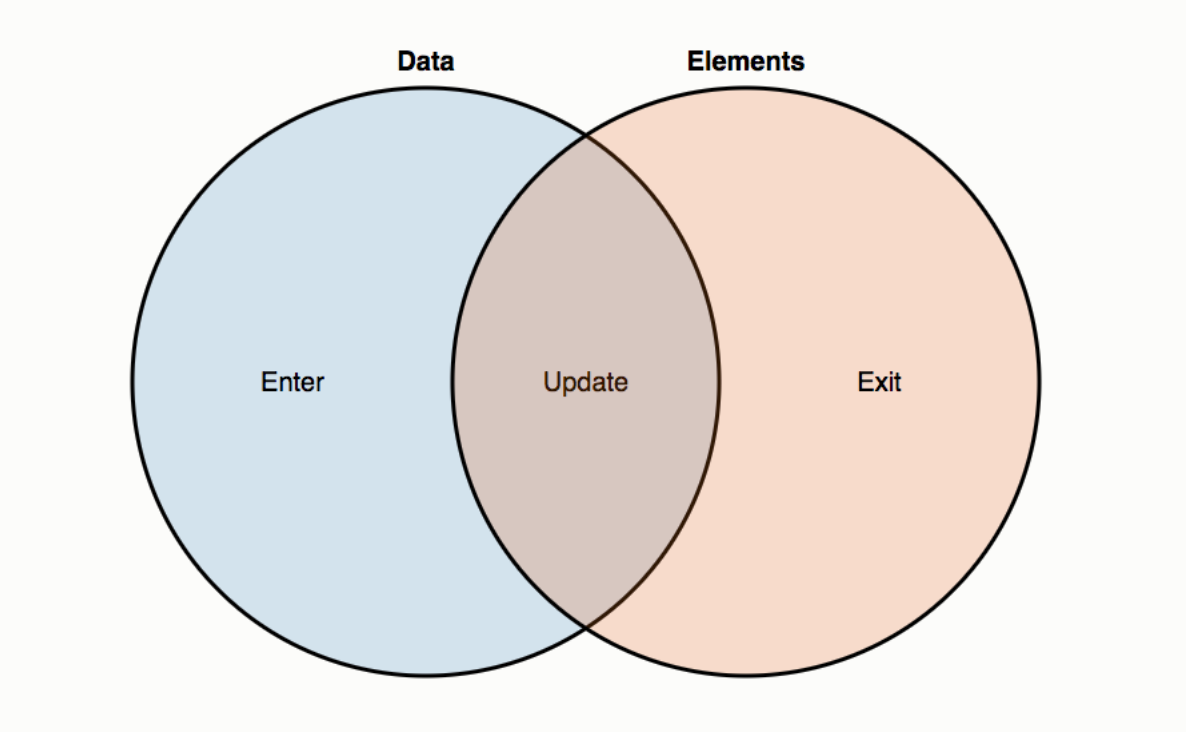
D3 Main Concepts

- DATA JOINS - Load a given dataset, then, for each of its elements, creates an SVG object with associated properties (shape, colors, values) and behaviors (transitions, events)

```
// Data
  var data = [
     { name:"Ireland",  income:53000, life: 78, pop:6378, color: "green"},
     { name:"Norway",   income:73000, life: 87, pop:5084, color: "blue" },
     { name:"Tanzania", income:27000, life: 50, pop:3407, color: "grey" }
  ];
// Create SVG container
  var svg = d3.select("#hook").append("svg")
        .attr("width", 120)
        .attr("height", 120)
        .style("background-color", "#D0D0D0");
// Create SVG elements from data
    svg.selectAll("circle")                 // create virtual circle template
      .data(data)                           // bind data
    .enter()                                // for each row in data...
      .append("circle")                     // bind circle & data row such that...
        .attr("id", function(d) { return d.name })          // set the circle's id according to
        .attr("cx", function(d) { return d.income /1000  }) // set the circle's horizontal posi
        .attr("cy", function(d) { return d.life })          // set the circle's vertical positi
        .attr("r",  function(d) { return d.pop /1000 *2 })  // set the circle's radius accordin
        .attr("fill",function(d){ return d.color });        // set the circle's color according
```

D3 Main Concepts

- ENTER, UPDATE and EXIT SELECTIONS
  - Best done by the author Mike Bostock - Thinking with Joins
  - Also D3 In Depth book (in progress) is great!



D3 - Other concepts

- Lots of convenience functions for loading and manipulating data, working with arrays
- Axes, Scales, Colors
- Manipulate structure of your data - d3.nest()
- D3 API Reference
- Writing reusable components
  - A good place to start is by reading Towards Resuable Charts where Mike Bostock proposes a convention for creating reusable charts.
  - This blog post and this book do a great job on clarifying the resuable charts api with examples and explanations.

Again, what is D3?

- Has everything you need for visualizing complex data BUT you will not find commands like barchart, scatterplot, or piechart, or even map
- Builds vizualizations (and other content) from its basic HTML5 or SVG elements (e.g. < g> , , < r e c t > , < p a th >).
- Most and foremost a DOM manipulation library (in this regard similar to jquery).
- Provides handy utilities for processing data (array, time series, geo data)
- Comes with a lot of functions and methods to create mapping functions that will map your data directly to properties on html elements
- Filled with algorithms (voronoi, quadtrees, circle fitting, convex hull, projections)

"One of the most interesting aspects about d3 is that it is the intersection between design and code, math and art, data and story." Ian Johnson

Review D3 Code

Let's look at the d3 code in `final` and step through some of it.

Other options

- Other data visualization libraries
- Rickshaw, Highcharts, NVD3 are libraries built on top of D3
- D3 and Crossfilter (Fast Multidimensional Filtering for Coordinated Views)
- Open-source tools binding D3 to R, Python
- Vega - higher-level visualization specification language on top of D3