# The First D3 Homework

## Starting with D3.js

**Overview**

Get started with D3.js. It is recommended that you read through the examples at https://github.com/mbostock/d3/wiki/Tutorials

Start at the beginning, and go down the list (you can probably skip the extra Korean tutorial). I recommend working through the list until “General Update Pattern, III”. You can use Google and look at other examples to help you with the assignment.

Start with the example D3.js project (“simple_example”).

**Instructions**

1) Set up Node.js and run a simple local web server:
(instructions modified from http://www.jhh.me/blog/2012/12/24/setting-up-http-server-on-windows-with-node-js/ )
    a. It's easy to setup Node.js and install NPM apps on Windows systems. Here’s a guide how to setup a HTTP server. No programming skills required! (Basically it’s the same process for any OS!)

    b. First you must install NodeJS: https://nodejs.org/

    c. Open the command prompt to run following commands.
        i. Windows 7 or 8: Open Start and write cmd into the search and press enter.
        ii. Windows XP: Open Start and select Run and write cmd and press enter.
        iii. Mac OS X: Open a terminal (Command+space to bring up spotlight. Type "Terminal" and press enter)

    d. Run this command to install a HTTP server: `npm install http-server -g`

    e. Run this command to start the HTTP server: `http-server path`
    (where "path" is the path to the root folder with your html (index.html). Your folder must have some files before it works. Use the “simple_example” to try it out)

    f. Open http://localhost:8080/ in your browser to start the application. Use the path for the directory of “simple_example” (extracted) in step “e”, and then go to http://localhost:8080

2) Edit the files of simple_example to complete the following instructions. You should mostly (or only) need to edit “main.js” from “simple_example\scripts”.

    a. **Blue dots!**
    Main.js is loading a file called “stream_1.json” and creating a scatterplot. Make the dots blue instead of black.
    b. **Add green squares!**
    Keep the blue dots from stream_1.json, and **also** load the data from stream_2.json (both data files are in the data directory). Make tiny green squares for the data from stream_2.json.
    c. **Color changing!**
    Add `mouseover` and `mouseout` events to both the green squares and the blue dots. Make

the shapes turn red when the mouse is over, and make it turn back to the correct original color when you mouse out.

d. **Brush and link!** (extra credit)
The data values from stream_1 and stream_2 have x and y values (xVal and yVal). Notice that the x values in the two files are the same. When you click on shape, make the other shape with the same xVal turn yellow.

**What you turn in:**

1. Your code. Zip your files (keep the directory organization).
2. A pdf file that includes:
    a) Screen shots of the visualization. These should match what your code generates.
    b) Reflection points. List the top three things that were difficult or confusing when you worked on this assignment.

**Grading:**
Graded out of 10 points (but 13 points are possible).

+5 points for getting it working and making the dots blue

+3 points for adding the green squares

+2 points for color changing (changing to red and back)

+3 points for brushing and linking (extra credit)