hacking the mooc-roscope

getting set up for hacking mooc visualisations

introduction ::

d3 is a data-oriented Web micro-framework that allows Web programmers to focus on data rather than everything around it. It follows the tradition of jQuery to provide 'fluent interfaces' that makes code extremely succinct and (sometimes!) easy to read.

Unlike using Matlab, R, SPSS or Excel, however, it is a general-purpose *data drawing* API*.* This means two things: first, that it takes a bit more time and investment to build things in d3 than it does to plot things in any high level tool or statistical analysis toolkit. But it also means that you have far greater flexibility about what you can build - in fact, you can probably build almost any kind of data visualisation you can think of with the right amount of d3.

This tutorial will walk through the basics of getting started with a web development environment so you can create visualisations in d3.

prerequisites & background reading ::

This tutorial assumes a familiarity with Web technologies, in particular with prior experience building static Web sties. Proficiency with JS is recommended but not absolutely necessary; a rudimentary knowledge of JS syntax, however, is recommended. I recommend "Eloquent Javascript" (http://eloquentjavascript.net/contents.html) for beginners, and "Javascript: The Good Parts" for advanced JS programmers.

Last & perhaps most importantly, you will need to learn d3. There are many great Web resources for learning d3 at the end of this tutorial.

Part 1 : Getting Set up

Let's set up your development environment. If you've done lots of web development before, you probably already have everything you need. Since we are going to collaborate through github, we are going to need three main things:

1a. Version control: **git**
1b. Library management: **node/npm**
1c**.** A local web server

1a. Installing git

Mac (GUI) - If you'd prefer a graphical client, install GitHub for Mac here:
        https://mac.github.com/

Mac (command line): Install Mac OS X **Developer Tools** (a free download through the App Store). This is a lot of stuff, so if you'd prefer not to do this, you can install git **homebrew** as follows:
        brew install git:
        if you don't have homebrew, get it at **http://brew.sh/**

Linux: You probably already have git. Type 'git' to see. If not, check your distribution-specific instructions for installing it. On Ubuntu you can type apt-get install git

Windows: Download GitHub for Windows: **https://windows.github.com/**

1b. Installing node/npm library management tools

Web developers these days use **npm,** the node package manager, which comes with NodeJS to manage their libraries. So we simply install NodeJS

Mac (if you have homebrew): brew install node
Mac (if you don't) : Go to https://nodejs.org/download/ and download the appropriate pcakge

Linux: sudo apt-get install node

Windows: Go to https://nodejs.org/download/ and install it manually

Once you have NodeJS installed, you can install bower / gulp etc

All platforms: npm install -g bower

1c. A local web server

Now that you have the hard stuff done, installing everything else you will potentially ever need for Web development is easy: Simply use npm.

To get a web server, open the command line and type :

> npm install -g http-server

That should work on all platforms. Later on, you can do the same for very useful utilities such as **gulp** (a 'Makefile' for js apps), **jshint** (a syntax/semantics checker for JS, etc).

Part 2 : Checking out a Project

Next let's check out an example project. Open your browser and go to github.com

> https://github.com/sociam/mooc-dendro1

Now, click the fork button in the upper right. **You may need to create a GitHub account and log in if you haven't already.**

Forking a project creates a separate copy of the entire repository owned by you. You can then clone the repository, make changes, and commit them as you please.

To clone your own repository after you've forked it:

From your forked repository page, copy the **repository URL** in the lower right.
If you're using the GitHub GUI: Open it and select Clone Repository, and paste in your repository URL.

If you're using git the command line tool, type

> **git clone** <URL to your repository>

for example, for my fork I would do: **git clone git@github.com:electronicmax/mooc-dendro1.git** but you should use your own username

Part 3: Running it.

Now you have a working project directory where you can run your project, edit it, create new ones, and commit them to a safe place.

Let's first try running the project. For this, you will need a copy of a mooc dataset. Ask the hackathon organisers for URLs for the mooc datasets. Note that we can't check them into the repository because we don't have permission to share the data with the world, and all github repositories are public.

3a. Start with the portus_comments_clean220714.csv dataset. Get the URL from the organisers, open it in a browser and save the file under the **data** subdirectory.

3b. Next, install your javascript libraries. We are going to use **d3**, **jQuery** and a few other very useful libraries in our project. The list of libraries the example project needs are declared inside the file **bower.json** in the project working directory. If you want to use additional libraries, put them in there.

To install all the project libraries listed in bower.json, simply type **bower install** This should populate the directory called lib/ in your project directory with the latest versions of each of the libraries you listed in bower.json.

3c. Start yourweb server.   From the project directory, type:

      http-server

It should tell you that it's hosting the directory at http://localhost:8080 That's a hint to open a browser to that directory. Open your browser (I recommend Google Chrome) to the first example at http://localhost:8080/dendro1.html You should see a simple dendrogram visualisation appear.

If it doesn't, something's wrong. Open up the Javascript Console by going to the View menu, selecting Developer, and going to Javascript Console. You should see all the debug output and errors there.

Part 4: Making changes

Now you're ready to make your own. The main code for the example is in js/ dendro1.js - which specifies the URL of the data file it loads, and the code for drawing the graph. You can now make your own visualisations by diving in and editing this code.

Find examples at the D3 Gallery here:
      https://github.com/mbostock/d3/wiki/Gallery

Once you have made changes and want to put them up on GitHub, make sure you **commit** your changes and **push** them to your repository.

For resource for learning d3, try these:

      Official D3 Tutorials: https://github.com/mbostock/d3/wiki/Tutorials
      O'Reilly Interactive Data Vis : http://alignedleft.com/tutorials/d3
      Dashing D3: https://www.dashingd3js.com/table-of-contents

Ask the hackathon organisers about how to best share your code with others and have fun!