

Princeton Hack Classes

Web Development

Week #1

Collin Stedman
cstedman@princeton.edu
Frist Center 008

Contents:

1. Environment Explanation
2. OSX Setup Instructions
3. Windows Setup Instructions

Environment Explanation

If I asked you to write an essay to be submitted electronically, how would you complete the assignment? Obviously, you would need a way to write text to a document, probably Microsoft Word or its equivalent, or maybe a Google Doc (which I am using now). If you only had a very old machine, you might write in Notepad or TextEdit. An experienced programmer might even choose to write in Vim. At some point during the writing process, you might need to look up a word or use a thesaurus. Your computer might have an app for that, or you could just search the web. You would probably want to save from time to time, perhaps to a USB or your Dropbox.

Even the simplest of exercises require you to make use of many different tools. You probably have a go-to set of tools for the tasks you perform the most often. It's possible that you're so used to using one tool for a particular job that you aren't even aware of alternatives. Mastering new tools takes time, and switching between tools can be frustrating. If your experience in the world of web development is anything like my own, you may find that programmers' tools present one of the largest learning curves you must overcome. There are many of them, they are frequently complex, and you generally won't have the luxury of learning them one at a time.

I believe that many of the people who try and fail to teach themselves to program do so because they are overwhelmed by tools. I was one of those people, in fact. I tried to teach myself C++ in high school, but I was mystified by terminals, compilers, editors, and debuggers. I find it ironic that people should encounter so many difficulties with software intended to make the process easier.

It all sounds discouraging, but it comes with the territory. Everybody, professional programmers included, must continually learn tools which change and break and have textbooks for manuals. It's no surprise that programmers take such pride in the simplicity and efficiency of their development environments. Environment configuration is an art.

There are three tools which are essential to web development:

1. The browser. The applications you make are almost certainly meant to be seen on a browser, and you would obviously like to test them out before deploying them. You can do much more than simply view pages, however. Many browsers provide developer tools which allow you to view network requests, jump between page elements and their HTML representations, and even change styles on the fly. You owe it to yourself to use a browser which offers such tools.
2. The editor. Editors are a programmer's version of Microsoft Word. Fundamentally, they open, modify, and save files. Of course, the differences are in the details. Since programmers spend a huge amount of time writing code in editors, they are designed for extreme efficiency. Syntax highlighting, auto-completion, and fast file navigation are just a few of the popular features provided by editors. Most editors are highly customizable, allowing programmers to configure them for their particular needs. The particular features and customization options differ from editor to editor. Some of the most thoroughly featured editors are termed "integrated development environments", or IDEs.
3. The terminal/shell. Many of you have probably seen images from movies like the Matrix of long lines of green text on an otherwise black computer screen. These images are inspired by the terminal, a tool which serves as the GUI (graphical user interface) for a shell. A shell, in turn, allows you to interact directly with your operating system through text commands. Shells allow you to navigate through your file system and run commands, much as you would if you were using a mouse. However, the text-based approach makes it easy to automate processes by creating "scripts" and run programs in different modes through arguments on the "command line". Terminals and shells both provide a large number of customization options.

So there's no getting around it. We're going to have to use real tools in this class, and in the long run you'll appreciate that we did. I'm always happy to help people who want to improve or customize their development environment, so please do not hesitate to ask me questions. There are many options out there, and you may find that you prefer deviating somewhat from my instructions here. That said, here are my recommendations:

OSX Setup Instructions

One of the greatest advantages of OSX over Windows is the large selection of terminals and shells. All Macs come with a very decent default terminal running a "bash" shell. For most of you, this will be a fine option, and you need not download anything else. If you are ambitious, however, I highly recommend a replacement terminal called iTerm, which provides more customizable features than the default terminal. You can find that here:

<http://www.iterm2.com/#/section/home>.

You almost certainly don't need to change the default shell. bash is an industry standard. That being said, there are other options out there. I personally use a shell called zsh. Talk to me if

you're interested.

Another advantage of OSX over Windows is Safari as the default browser. Unlike (most versions of) Internet Explorer, Safari offers full-featured developer tools which you can activate by following these instructions:

https://developer.apple.com/library/safari/documentation/AppleApplications/Conceptual/Safari_Developer_Guide/GettingStarted/GettingStarted.html#//apple_ref/doc/uid/TP40007874-CH2-SW2

That being said, I still prefer Chrome or Firefox. Their developer tools are always available by right-clicking and going to "Inspect Element" (or by pressing "command option i"). Firefox also has a popular free upgrade to their tools called Firebug. I don't know too much about it, so I'm not going to provide the link here, but you can find it easily on Google. I'll use Chrome in class, so I recommend you do the same unless you have a strong personal preference.

Finally, you need an editor. As I said before, there are many options out there. I've probably gone into too much detail already, so I'm going to cop out and strongly recommend Sublime Text, which you can download for free here:

<http://www.sublimetext.com/3>

Sublime Text has many powerful features and is very user friendly. It touts auto-completion and syntax highlighting for most languages right out the box. It's also quite a bit prettier than most other editors. A huge number of programmers now use Sublime as their default editor, and I recommend you do the same unless you understand the implications of making a different choice. Come talk to me if you're interested in the alternatives.

And no, I don't use Sublime. I use Vim. It doesn't make any difference for this class.

Windows Setup Instructions

Everything I said about browsers and editors for OSX still applies for Windows. I recommend you use Chrome or Firefox if you aren't already. Really, don't use Internet Explorer for web development. There are two exceptions:

1. You're checking to see how your site renders on Internet Explorer. That's fair.
2. You're using IE 11, which I hear has good developer tools. If you're using IE 11, I'd love to talk to you.

The big difference between Windows and OSX is the support for terminals and Unix-like shells. Unfortunately, Windows doesn't come with a good terminal, and its operating system is based on DOS instead of Unix. What does that mean? To put it simply, you may run into

compatibility problems many standard programming tools are designed for a different environment. While there is no perfect solution to this problem, you can avoid many issues by downloading a terminal/shell that simulates a Unix experience. I recommend the “git bash” shell, which is really just a complementary shell installed on your machine when you download git, a popular version control tool. Why the weird recommendation? The bash shell installed is really just msys, but the installation is easier. If you don’t know what I’m talking about, don’t worry about it. Anyway, here’s the link:

<http://msysgit.github.io/>