Development Environments

Objectives, You Should Learn How To:

- Describe the differences between Production, staging and development environments.
- Setup a basic development environment with:
  - Chrome - Modern web browser (IE and Safari don't count).
  - Sublime Text 3 - text editor.
  - Macdown - a "markdown" editor.
  - DiffMerge - a "difftool".
  - Homebrew - package manager.
  - Git - version control.
- Briefly describe the purpose of each of the above.

Content

Production, Staging, and Development

In most web software companies there are 3 major "environments" which host and run the companies software applications: production, staging, and development.

Production is a server (or collection of servers) that is live, on the open web, and accessible to the public. When you type www.google.com into your browser, you are making a request to Google's "production environment". Production environments are carefully configured and designed process users requests as quickly as possible.

Staging is a typically a clone of the production environment. The purpose of staging is to allow developers to test their code somewhere that behaves identically to production, but isn't available to the public. These are often available within the companies local network, or gated behind a user/password authentication screen.

Your development environment is your laptop, desktop, or whatever device you use to write code.

The Purpose of a Development Environment

Unlike production and staging, this environment is optimized for humans, instead of machines. Where production and staging need to respond to requests as fast as possible, development environments should help you generate code as quickly as possible.

Creating a good development environment is a crucial step for every software engineer. There are a number of tools that you'll use every day to help you read, write, maintain, run, and navigate source code.

Picking good tools is just the start. We'll also be introducing a number of useful keyboard shortcuts and techniques. The goal of a good development environment is to maximize time spent thinking about the logic of your code, and minimize time spent worrying about how to write it.

Essential Tools

We have chosen an essential toolset for the Galvanize FS program. Reasonable people often disagree about the perfect tool for any given task, but we can provide more support if you use the same tools we do.

There are 3 essential tasks that every engineer must always do. We've chosen a basic toolset to help us with these tasks:

- Editing code.
  - Sublime Text 3: for editing source code and text (HTML/CSS/JavaScript/Ruby/...)
  - MacDown: for editing "markdown" a language used for documentation.
- Run code. Engineers are constantly testing and using their apps
  - Chrome: a good webbrowser for running and debugging websites.
  - The Command Line: for running servers, scripts, and so much more.
- Commit code to the master codebase. when the code works, it has to be saved.
  - Git: a "version control" system that helps devs collaborate and save code.

Finally, we're also going to have you install Homebrew, a "package manager". Frequently developers will need to install another tool or library. Package managers makes installing software much easier. Because it makes everything else so much easier, we're going to start with Homebrew.

Excercises - Install the Following Tools

Homebrew

What is it? What will I be using it for?

Homebrew is a package manager (a tool that helps us install additional technologies/tools) for OSX. Throughout the course we'll be installing tools to help us with specific tasks, such as Node for run Javascript servers; or Mongo and Postgres as databases. Homebrew will significantly improve your speed and accuracy when installing these "packages" of software.

How do I install it?

In terminal, run:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Check your Hombrew install with:

```
which brew
```

Sublime Text 3 (subl alias) / Package Manager

What is it? What will I be using it for?

Sublime Text is going to be our editor of choice. There are literally hundreds of text editors to choose from but we **strongly** recommend Sublime Text 3. Sublime text is designed targeting for software engineering, and is very popular in the web development community.

I have Sublime Text 2....that works right?

Unfortunately not :(- although it is still in beta, Sublime Text 3 has a set of packages (we'll discuss later what those are) which are incredibly useful and are not available in Sublime Text 2. If you already have Sublime Text 2, please delete it and install Sublime Text 3.

How do I install it?

http://www.sublimetext.com/3

Using the subl alias

It would be really nice if we could open up sublime from the terminal. To get started with that read this (we will go through this process together):

Please read this Stack Overflow question before we do this together.

NEXT STEPS

While Sublime Text 3 comes with some nice things out of the box, there are a bunch of additional tools we will want to add to our text editor. These extra tools can do cool things like tell us if we've written a line of code incorrectly - think spelling and grammar check for JavaScript.

Sublime Text has a "package manager" of it's own. This is similar to Homebrew, but specific to Sublime Text packages. Head over tohttps://packagecontrol.io/installation and follow the instructions for Sublime Text 3. If you install this correctly, you should be able to type `command+shift+p` in sublime and type in `install package` and see something.

MacDown

What is it? What will I be using it for?

When we write readme files for our repositories, the language we use is Markdown, which is a text to HTML converter (you can read more about it herehttp://daringfireball.net/projects/markdown/). It's essential to understand how Markdown works as well as have a tool where we can write Markdown. Sublime Text 3 can help us with that, but there are better tools - MacDown is one of them!

How do I install it?

http://macdown.uranusjr.com/

DiffMerge

What is it? What will I be using it for?

DiffMerge is a "diff tool". It's built on top of a common unix command called `diff`, and it allows you to see the difference between two text files. It is very useful in conjunction with Git in order to see the difference between what you're currently working on, and the last time you saved the code.

How do I install it?

https://sourcegear.com/diffmerge/downloads.php

Chrome

What is it? What will I be using it for?

Chrome is a browser that not only offers some useful features for browsing, but it is a **MUST HAVE** for developing and debugging HTML, CSS, and JavaScript. PLEASE make chrome your default browser - we will be using it almost every day.

Some browsers (IE and Safari, especially old versions) do not play nice with many modern web standards. Having a modern web browser is crucial for an efficent development environment.

How do I install it?

https://www.google.com/chrome/browser/desktop/

Git

What is it? What will I be using it for?

Git is a version control system that we will be using to keep track of changes in our code. It is a **MUST** know for any developer and we will be spending lots of time on it over the course.

Version control allows programmers to make changes to their code without breaking the "master" copy; roll back changes that didn't work; backup their work across many "branches"; collaborate with other programmers on the same codebase; and so much more.

There are literally no successful software companies that do not use some kind of version control.

How do I install it?

In the terminal, run:

```
brew update
```

```
brew install git
```

Essential Shortcuts

Developers tend to become "power users" in order to efficently move through and between the various tools they use regularly. This is a starting cheat sheet for commands that can help you navigate and use your new development environment more effectively.

OSX

`command + tab` : switch open applications

`command + `` : (when already in command + tab) switch open applications in reverse

`command + `` : goes between different windows of current program

`command + q` (while in open applications) : quit an application

Chrome / Sublime / Bash / most apps with tabs

`command + n` : create new window (`shift + command + n` for Sublime)

`command + t` : create new tab (`command + n` for Sublime)

`command + w` : close tab or window if only one tab

`shift + command + w` : close window with multiple open tabs

`command + q` : quit application

`command + shift + ]` : move right one tab

`command + shift + [` : move left one tab

`command + NUMBER` : move to tab number NUMBER (not in Terminal)

`fn + up` : page up

`fn + down` : page down

`alt + command + arrow keys` : move right or left tabs

Chrome

`command + r` : reload page

`command + option + j` : open the javascript console

`command + ]` : move forward through history

`command + [` : move backward through history

`command + shift + t` : reopen last closed tab (up to 10)

Sublime

`command + ctrl + f` : toggle fullscreen

`command + ctrl + shift + f` : toggle distraction free

`option + arrows` : move one word at a time

`command + arrows` : move to beginning and end of line

`shift + above commands` : select all the text

`delete + the above` : delete all the text

`fn + delete` : delete forward

`command + k` , command k : delete to end of line

`command + delete` : delete to beginning of line `fn + up / down` : page up / page down

`fn + right / left` : beginning and end of file

`command + z` : undo

`command + shift + z` : redo

`command + y` :redo

`command + /` : toggle comment for line

`command + ]` : shift indenting right

`command + [` : shift indenting left

`command + l` : select the whole line

`command + c` (with selection) : copy selection

`command + x` (with selection) : cut selection

`command + v` : paste most recent cut or copy

`command + shift + v` : paste with proper indentation

`command + x` : cut the whole line

`command + c` : copy the whole line

`command + enter` : create new line below

`command + shift` + enter : create new line above

`command + shift + d` : copy current line below

`command + control + arrows` : move entire line up and down

`command + k, command + u` : make selection upper case

`command + k, command + l` : make selection lower case

`command + '` : select all within quotes

`control + m` : move to beginning/end of parens and braces

`command + d` : select word(s)

`command + k` while selecting words : don't select word

`command + ctrl + g` : select all of words

`option + mouseDrag` : column select

`command + mouseClick` : custom multiple cursor

`command + p` : goto anything (use with :line_number, @symbol, #term)

`command + r` : search all function names in current file `command + shift + p` : set syntax (and many many other things)

`command + f` : find

`command + shift + f` : massive find

`command + k, command + b` : toggle file bar

`command + option + NUMBER` : split screen

`control + shift + NUMBER` : move current file to pane

`command + shift + l` : multi line cursor based on selection

`command + control + up/down` : move a line up or down