Terminal and Editor

UNIX Essentials

If this is your first time using UNIX then you'll need a few of the most essential pieces to be able to complete your work:

- Open Terminal by typing `command-spacebar` to open Spotlight, then type `Terminal` and hit enter
- You now have a single terminal window. This window can open multiple tabs by typing `command-t`
- The prompt on the left tells you a bit about what folder you're currently in. But try typing `pwd` in the terminal and hit enter to print out your "present working directory"

Listing Files

- `ls` will list the files in the current folder
- `ls -lA` will list the files in the current folder along with a bunch of info about them

Working with Files

- Files that start with a `.`, like the `.bash_profile` you'll work with later, are hidden files. If you just use `ls` they won't show up, but `ls -lA` will show them.
- `touch` is used to create blank files. Try `touch sample_file` then `ls`.
- `rm` is used to remove files. Remove that sample with `rm sample_file`
- `which` tells you where on the file system a program is. Try `which ruby` to see the full path to your Ruby executable.

Working with Directories

- `mkdir` will make a directory. Go ahead and enter `mkdir sample_directory` to create a directory
- `cd` stands for "change directory". Enter `cd sample_directory` to move into your new directory
- The tilde (`~`) is a shortcut for your "home" directory. You can enter `cd ~` from any folder on the system and you'll jump back to your home directory.
- The single period (`.`) is a reference to the current directory. If you enter `cd .` it won't go anywhere. But the period is useful especially with Git which you'll see soon.
- The double period (`..`) is a reference to the parent directory of the current directory (one step up the tree). Try entering `cd ..` then `ls` and you should see your user folder. `cd` back into that.
- Removing directories is a bit different. Try `rm -rf sample_directory` to remove our previously created sample directory

Setting Up Terminal Access for Atom

One of the things you'll do frequently is open an entire folder (like when working on a project) in your text editor. Let's get that setup:

- Open Atom (`command-spacebar` for spotlight, type `Atom`, and hit enter).
- Click the `Atom` menu in the top left corner
- Click `Install Shell Commands`
- Return to your terminal and enter `which atom`. You should get back `/usr/local/bin/atom`
- Enter `atom .` to open your user directory in Atom.
- Experiment with creating a file in Atom and using `ls` in the terminal to see it. Try creating a file in the terminal with `touch` and see if it shows up in Atom.

Customizing Your Terminal

A little bit of increased efficiency in your use of the Unix environment and your editor can pay significant dividends over time. Let's experiment with customizing and adding to your terminal and editor.

- Open `~/.bash_profile` in a text editor (ex: `atom ~/.bash_profile`)
- You can check out dotfiles on GitHub to see how serious people get: http://dotfiles.github.com/ [http://dotfiles.github.com/]

The Essentials

- `export` to set environment variables
- `alias` for shorthand commands, like I define `e` to launch my editor
- `source` to run scripts of bash commands

Dotfiles

Snippets from my `.bash_profile` are below. The top three lines setup a yellow lightning bolt as my prompt because, well, it's awesome.

We have set up some dotfiles you can use, go to bootstrap new students [https://github.com/turingschool/bootstrap_new_students] and follow the instructions there.

```
export PS1="\W \[\033[0;33m\]⚡\[\033[0;39m\] "
export CLICOLOR=1
export LSCOLORS=ExFxBxDxCxegedabagacad

export EDITOR='/usr/local/bin/atom'
export CC=/usr/local/bin/gcc-4.2

# My general projects directory:
alias cdp="cd /Users/jcasimir/Dropbox/Projects/"

# My most commonly used project, "curriculum":
alias cdc="cd /Users/jcasimir/Dropbox/Projects/curriculum/source"

# Use "be" instead of "bundle exec" for Rails
alias be="bundle exec $1"

# Use "a" and a folder/file to launch Atom
alias a="atom $1"

# Enable git's tab-completion library
```

```
21  source /usr/local/etc/bash_completion.d/git-completion.bash
22
23  # shortcuts for git
24  alias ga="git add"
25  alias gb="git branch"
26  alias gd="git diff --patience --ignore-space-change"
27  alias gh="git log --pretty=format:\"%Cgreen%h%Creset %Cblue%ad%Crese
28  alias gc="git checkout"
29  alias gs="git status"
30
31  # programs that launch editors (e.g. git) will use Atom
32  export EDITOR="atom -w"
```

Customizing Atom

Check out the Atom docs that explain how to start customizing it:
https://atom.io/docs/latest/customizing-atom
[https://atom.io/docs/latest/customizing-atom]