**Command Line**

Why are we learning command line?

- The reason of why it is so hard to learn programming on your own is because of the sheer amount of technology out there to learn. Like all the different buzzwords, it can get a bit overwhelming. Knowing how to differentiate between React, Angular, Ruby on Rails, or identifying what tech is simply just a fad or something worth investing your time into.

-The command line is something for sure that I can tell you, is not going away. You will need to learn the command line. The better you know it the more efficient you will be at your job.

-The earlier it gets burned in, the better. Because that means you will have fifteen or so weeks to get used to it.

-A directory is a location for storing files on your computer Directories are found in a hierarchical file system (a folder within a folder within a folder…), such as in Linux, MS-DOS, OS/2 and Unix. We will be using the Unix command line installed in Mac.

Command Line Basics:

pwd – Print working directory

cd – change the working directory

cd .. – go back one directory

ls – list contents of the working directory

mkdir – make a new directory within the working directory

touch – create a new empty file

less – show the contents of the file

atom – open up the atom text editor

cp – copy a file or a directory

mv – move/rename a file or directory

rm – remove a file (or potentially a directory)

rmdir – remove a directory

clear – to clear the contents of your terminal window

Shortcuts that will make you more productive in your use of the terminal

up arrow – reproduce the previous command just executed

TAB – auto completion

cmd-tab – cycle through windows

cmd-s – save a file

cmd-space – bring up spotlight

cmd-+ - increase the size of text

cmd-- - decrease the size of text

Working with the terminal

You can bring up your terminal by using the command space shortcut and type in terminal then hit enter. The way to make you text size bigger is to use: command +

At any given time when you are working with the terminal, you are working within some directory. What this means that whatever commands that you are working with that has to deal with files or creating things, it will search for those files and folders starting from your working directory. Pwd will tell you what your current working directory is. Initially, typing pwd upon opening the terminal will tell you that you are starting at the home directory: /USERS/YOUR\_USERNAME

Enter your desktop directory by using: cd Desktop

Create a MyProjects folder using: mkdir MyProjects

Enter ls to see a the content in Desktop using: ls

You can also check your finder and look at the MyProjects folder in your desktop to see things being added in in real time.

If you do not see your Home folder on the Sidebar, goto menu: Finder -> Preferences. Click on the sidebar tab and then click on the house icon and then your home directory should appear on your finder sidebar

cd stands for change directory is changes your current working directory to whatever you want to change it to.

~ is an alias for your home directory, so if you do cd ~, it will take you to your home directory. If you do pwd, you will find that you are back at the beginning where you initially started from.

touch is a command that will access a file if it already exist but if it doesn’t exist then it will create that file for the first time with 0 data.

cd back into the MyProjects directory and to create a new file, use touch to create a new file called file1 using: touch file1

use cd .. to retrace back from one directory

to move a file, use mv [destination directory]

to change a file’s name from hello.py to hello.js: mv hello.py hello.js

rm –r will delete the entire directory and all the subdirectories involved inside that directory. –r stands for recursive. Quite dangerous tbh. It can even be done on your home directory! don’t ever do that ^^.

rm should be used to remove files

you can use spaces in file names, the file or folder name just needs to be surrounded in quotes “”.

**Python 101**

Basic Programming Concepts

Variables, Strings Numbers, Printing, User Input, String formatting, If statements, Booleans, while Loops.

Printing

print ‘Hello, world’

Macintosh HD:Users:Banette:Desktop:sort:Screen Shot 2017-06-09 at 2.30.38 PM.png

This is a hello world program in python. Hello world is a thing that programmers do to learn a learn language or framework. The word print is a statement and the second part is a string

Every print statement without a dangling comma, will create a new line each time it is called and displays in the terminal.

Strings

A string literal in python is created using double quotes or single quotes. If you use double quotes to wrap a string literal, you can use single quotes within it. If you really want to use single quotes you are able to escape it using a backslash. a moment where this may be useful is when your string looks like this:

‘I’m hungry’

this will fix it

‘I\’m hungry’

Python supports multiline strings and this can be done with either double quotes or single quotes.

Concatenating means to add something to an existing string. things you could add to a string is another string.

‘abc’ + ‘def’

to write python code straight on your terminal. Type python then press enter. then you can begin writing python that executes after each line.

Variables

How to declare a variable:

In declaring a variable you need two things: The value that you want to store and the name you want to call that value by.

name\_you\_want\_to\_call\_that\_value\_by = The\_value\_that\_you\_want\_to\_store

name = “Carl”

num = 24

Left hand side is like an identifier and the right hand side is a value.

a string can be a value along with several other datatypes like integers and others that will be mentioned later.

a variable can also store a list of number

Some datatypes

Datatypes

in addition to strings and numbers, you can store a list.

numbers = [1,4,2,5,6]

the contents in the right hand side is a list. the value of a variable can be a variety of different types of things.

Identifiers

the things that you can freely na≥?>me, like variable names, function names module names. However there are some limits as to what names you can give these things.

an identifier must start with a letter between a-z and A-Z (caps) or an \_ (underscore). It also cannot be a keyword. These are the list of reserved words that you cannot use as a identifier.

refer to Python101.key