OPEN TERMINAL

We will be doing some of our work in terminal during this course. If you don't know where to find it, you can use the Spotlight search tool (command-space) to find it by typing "terminal". Alternatively, Go to Applications/Utilities/, and you should see the Terminal program (black box icon). If you've never used Terminal or some kind of command line interface before, now is a good time to get started. Open "terminal" and poke around - this is a deep subject. For now, this is all you need to know to get your environment up and running. Check out the recommended book list for further reading.

For more advanced users, "iTerm2" provides more visual flexibility and has some nice features. Check it out here.

ALL AT ONCE VERSION

Open a terminal.

If you're really feeling confident, paste all the lines at once, but that might be hard to diagnose! Alternatively, paste these line by line. I

After each one completes, paste the next line. Watch out for line-wraps (e.g.: the first line is actually a wrap). Many of these commands will prompt you for your password, so you can't paste them as a block. Total run time is about fifteen minutes; the longest steps are the first two (installing brew, xcode, and python).

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew install python
curl -O https://bootstrap.pypa.io/get-pip.py
sudo -H python ./get-pip.py
echo -e '\n''export PATH=$HOME/Library/Python/2.7/bin:$PATH' >>
~/.bash_profile
pip install --user jupyter
pip install -U --user --force ipython
pip install --user matplotlib
brew install git
git clone https://github.com/calacademy-research/PythonLife.git
exit
```

Exit the terminal window (the ‘exit’ above should have done that) and re launch your terminal window. The step below you’ll execute every time you want to start the notebook.

Enter:

```
cd PythonLife
jupyter notebook
```

LONG VERSION

INSTALL HOMEBREW

Homebrew is a package manager for OS X. A package is a collection of code files that work together. Installing them usually means running a script (a bit of code) that puts certain files in the various directories. A lot of the packages you will want are going to have dependencies. That means they require you to have other packages already installed on your computer. Homebrew will find and install dependencies for you AND it will keep them organized in one location AND it can tell you when updates are available for them. On top of all of that it gives super helpful instructions when everything doesn't go smoothly. You can read more about it at Homebrew's website. For now, install Homebrew using the following line of code:

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

So what's going on here? The last bit is obviously a URL. If you were to open this URL in your browser, you would just see code. This is a Ruby script that tells your computer what to do to install Homebrew. The curl part is a command line tool that downloads files using URLs - it takes a URL and transfers whatever file it finds to your local disk. The -fsSL part is a combination of four option flags for curl that specify how to handle the file at the url. If you want to know more about what these flags do, type man curl at your command prompt. (You can use man in front of most commands to open up a manual page for that command.) We also need to actually execute this script (which is written in a language called “ruby”, which is a cousin to Python), so we used the command ruby at the beginning. The -e is an option flag for ruby that executes a string as one line of code. You may need to follow a few more instructions to finish the install, but Homebrew will help you do so.

INSTALL PYTHON

Let’s put that brew to work and get a nice recent copy of Python. You can use brew to install most programs that you’re accustomed to using on a linux server. Anytime you use Homebrew, you will start your command with brew followed by the Homebrew command you want to use. To install the latest version of python 2, type:

```
brew install python
```

This course is based on Python 2.7. While the language versions are pretty similar, Python 3 and Python 2.7 libraries are not compatible, and some important third party packages still only support 2.7.

If you want more flexibility in your python versions, check out pyenv. Pyenv will support multiple simultaneous versions of python simultaneously, and provides a simple command line tool to toggle between them. It is strongly recommended for users who are more comfortable with the command line.

INSTALL PIP

There are a few package managers that are specific to Python, and pip is the preferred one. The name pip stands for "pip installs packages". Fun fact - GNU, a family of open source UNIX utilities, stands for ‘Gnu’s not UNIX’. Recursion is an advanced topic that is beyond the scope of this course, but it’s a powerful tool. In any case, pip can be easily installed with python scripts that are available on the web. We can use curl, just like we did to get Homebrew.

```
curl -O https://bootstrap.pypa.io/get-pip.py
sudo -H python ./get-pip.py
```

This time we are getting and executing each script in two commands, where we did it all in one command before. Remember that you can look up what -O does with $ man curl, if you're curious.

It's possible that you will run into a permission issue here. Every file on your computer stores information about who can access and modify it. The get-pip.py script is going to try to write files to some of your system directories and it's possible that your user account doesn't have the right permissions. You can get around that though. If you get an error for one of these Python commands about permissions, type sudo before the rest of the command. Sudo stands for "superuser do". The superuser does have permission to modify system files and when you say sudo, you are acting as the superuser. You will need the admin password to do this.

For more information about using pip, you can go here.

INSTALL JUPYTER NOTEBOOK AND FRIENDS

This one's easy:

```
pip install --user jupyter
pip install -U --user --force ipython
pip install --user matplotlib
brew install git
```

Once you're done, everything has been installed locally. But your computer doesn't know where it is, yet. Let's tell it by adding to $PATH. $PATH tells your shell (bash) (aka: The thing that responds when you type in terminal) where to look to run programs. In geek speak, we call these "binaries".

```
echo -e '\n''export PATH=$HOME/Library/Python/2.7/bin:$PATH' >>
~/.bash_profile
```

Now that we've created the .bash_profile configuration file, let's restart the terminal. Close it and re-open it.

DOWNLOAD AND UNPACK THE CORE COURSE FILES

In terminal, change to the directory you'd like to work in. If you don't know how to do that from terminal, it's okay to do it right where you are, in "home". Type:

```
git clone https://github.com/calacademy-research/PythonLife.git
```

RUN JUPYTER NOTEBOOK

From the command line, go to your newly downloaded PythonLife folder and run notebook! The step below you'll execute every time you want to start the notebook.

```
cd PythonLife
jupyter notebook
```

This will launch the class in your browser. Check out the notebook basics!