- Home
- About
- Forum
- Locations
- Blog
- Code of Conduct
- Resources
- Contact
- RSS

# Get Your Mac Ready for Python Programming

**A PDX PyLady**

## March 06, 2013 | Tag: » absolute beginner, getting started, Mac OS X

Tweet

As an absolute beginner to programming, you can very quickly learn to do lots of cool things using just your Python interpreter and simple Python scripts. However, you're going to want a few more tools eventually to help you expand to more complex projects. Why not start getting familiar with them now?

For me, the hardest thing about branching out was figuring out what tools I really needed and how to interpret their install instructions. Here's a list of tools you will need and a list of easy steps to follow to get set up on a Mac. I'll explain what each line of code is doing so you can learn a little bit about working with a command line interface in the process. All of this should work on Mac OS X 10.7 and 10.8.

### 1. Install Xcode

Xcode is Apple's Integrated Development Environment (IDE), and there are some tools that come with it that we'll need later. You can get Xcode in the Apple appstore. It's a pretty big download, but this is the easiest way to get these tools on your machine. If you get bored while you're waiting, you could skip ahead and do steps 2, 5, and 6 now. If you want an alternative way to just get the tools you need, without the whole IDE, look here.

Once Xcode is installed, **you still need to install the Apple command line tools**! They're easy to get, though. Under the Xcode menu, click preferences. On the window that pops up, go to the Downloads tab. Find "command lines tools" and click the install button.

### 2. Open Terminal. Get Comfy.

If you've never used Terminal or some kind of command line interface before, it's a good idea to take a minute to familiarize yourself with how they work. Here's a quick intro. Later you you can learn a bit more with the free online book, Learn Command Line Interface the Hard Way.

I like to have a central place to store all of my programming projects, so I have a folder in my home directory called Code. To make a new folder called Code, open up Terminal. You should be in your home directory. Just to make sure you could type the following in your Terminal window. Don't type the $, though.

```
$ cd
```

Command line instructions usually start with $, which represents the end of your prompt. The prompt is the string of characters Terminal prints out to let you know it's ready to accept commands. The `cd` means change directory. If you don't specify where to go, it sends you to your home directory. Now, to make a folder called Code, you would type:

```
$ mkdir Code
```

The `mkdir` part means 'make directory'. A directory is analogous to what Finder calls folders. If you make a directory through your command line, it will show up as a folder in Finder. The `Code` part is an *argument* for this command. `mkdir` needs us to specify a string that will be the name of the new directory, so we pass it the `Code` argument. Unlike `cd`, `mkdir` will give us an error if we don't also provide a name. As shown below, the error message (second line) tells us that we used `mkdir` wrong and gives us some guidance on how to use it. Anything in brackets is optional, but the directory name is not!

```
$ mkdir
usage: mkdir [-pv] [-m mode] directory ...
```

**3. 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:

```
 $ ruby -e "$(curl -fsSL https://raw.github.com/mxcl/homebrew/go)"
```

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 transfers files using URLs. 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 Ruby script, 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, in this case, the `"$(curl â€¦ /go)"` part. You may need to follow a few more instructions to finish the install, but Homebrew will help you do so.

**4. Install Python**

Python comes with OS X, so you can probably don't need to do this step. You can check this by typing `python --version` into Terminal. If you get an error message, you need to install Python. If Terminal prints something like `Python 2.7.3` where the exact numbers you see may be different, you're all set to move on to step #5.

If for some reason you don't have Python or if you want to get the current version, you can now do this easily with Homebrew! 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, simply type:

```
$ brew install python
```

If you'd rather install the latest version of python 3, replace `python` with `python3`.

**5. 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". pip has one dependeny--distribute, but Homebrew doesn't know how to install either one. Luckily, both distribute and 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 http://python-distribute.org/distribute_setup.py
$ python distribute_setup.py
$ curl -O https://raw.github.com/pypa/pip/master/contrib/get-pip.py
$ 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.

### 6. Install virtualenv

A virtual environment is useful when you start to get invovled with projects that have different or conflicting dependencies that you dont want to install globally on your machine. For example, you might use a library that requires Python 3 and have another project that is only compatible with Python 2.

To install virtualenv, simply:

```
$ pip install virtualenv
```

Here is virtualenv's documentation.

### 7. Install Git and make a Github account

Git is a version control system. It keeps track of the revisions you make to your code and other files associated with a project, without storing multiple copies of each file. It will also help you merge your work with that of other programmers if you're working on a collaborative project. To install Git, type the following at your command prompt.

```
$ brew install git
```

Github is a website that uses git and is the most common way people share their code. You can use it just to back up your own projects or to have a centralized repository for a collaborative project, but it's also sort of a social media website for programmers. You can look at other people's code, follow projects you're interested in, submit bugs, and even contibute code to open source projects. If you want to get involved with an open source project or find good sample code, Github is the best way to do it. Go to github.com and follow the instructions to make your account.

If you're a little confused about this step, or want to know why and how you should use Github, you might want to read this.

### 8. What now?

You have a lot of powerful tools on your machine, now and it may be a little overwhelming. That's okay, you don't have to learn them all at once, but at least you have them set up now. but there are also plenty of resources freely available on the web to help you learn. Try browsing the Pyladies Resources page, or just using Google to find what you need. Talking with a more experienced programmer is always helpful as well. Have fun!

Keep up with what's going in the PyLadies community on our low-volume announcement list. Men and women from everywhere are welcome. Unsubscribe anytime. email address

Subscribe

Sponsor Us

Get Swag
Buy Stickers

## Upcoming Meetups

- Sorry, we are unable to reach Meetup.com at this time

## Recent Tweets

### PyLadies Locations

A Twitter list by @pyladies

List of all PyLadies locations!

PyLadies Berlin Retweeted

**Ellen König**
@ellen_koenig

I love @omosolatweets career development principle: „My career path has been to follow my nose. I always asked myself: What do I want to learn next?" — heard at her AMA at @PyLadiesBer

23m

**PyLadies Berlin**
@PyLadiesBer

lightning talk about #pandas from @tvasi about copies versus views in dataframes and .loc

Then our friends at @thecodepub invite us all to join their #meetup

"We believe competence has no gender - the future is equal & we are creating it"meetu.ps/c/2XT8M/CQlFh/d#community 🐍

Embed View on Twitter