

# Your First Steps with Python: Part I – Running your First Python Code

For the beginning coder, every step involved in getting up and running the programming language presents a new (and potentially frustrating) learning experience. In a [prior post](#) I gave an overview of my preferred Python distributions and my general thoughts on how to approach learning Python. In this and subsequent posts I will walk you through some of the concrete first steps involved in actually using Python.

I am assuming you are using a Mac (slightly different instructions for PC or Linux) and that you are interested in developing code especially for processing social media data. Most of the steps are generalizable to other applications, but I'm tailoring the instructions to those who wish to run my code.

## **Step One: Download and Install Python**

For Unix, Windows, and Mac users alike I recommend you install [Anaconda Python 2.7](#). This distribution of Python is free and easy to install. Moreover, it includes most of the add-on packages necessary for scientific computing, including Numpy, Pandas, iPython, Statsmodels, SQLAlchemy, and Matplotlib.

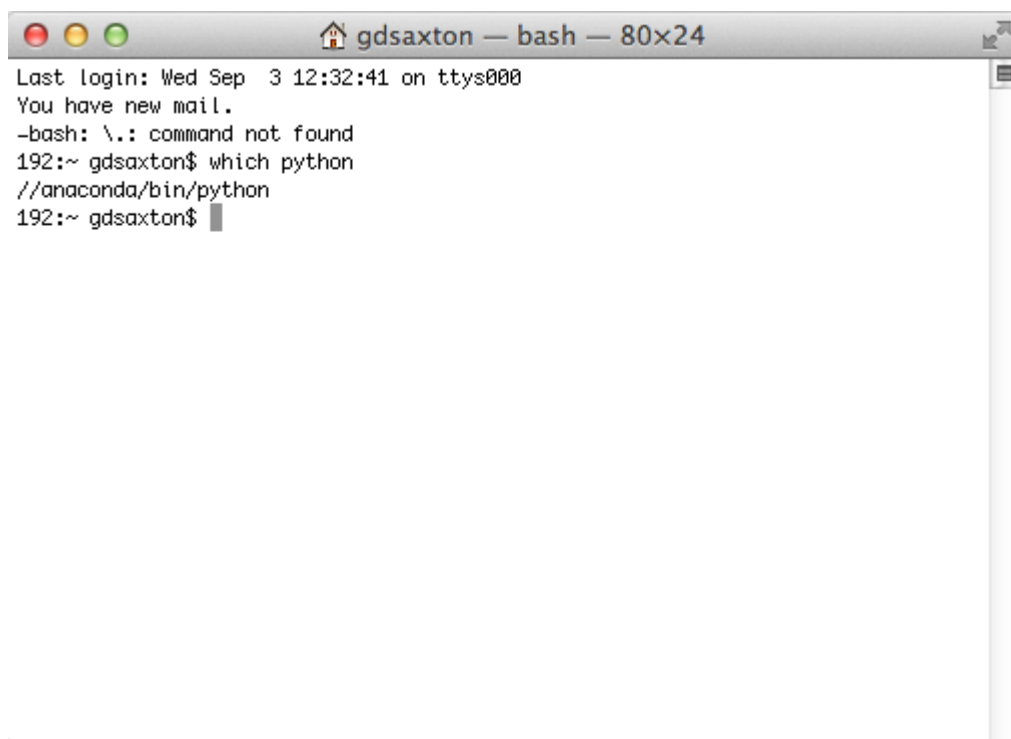
So, download the software. Double-click on it and run the software. Easy as that!

## **Step Two: Verify Anaconda is Now Default Version of Python**

To use Python you will occasionally have to do something in the [Terminal](#). The Terminal provides a text-based interface to

your computer's operating system and is necessary for certain permissions changes, etc. If this is your first programming experience you may not even know it's there. Now that you know, you may wish to you keep Terminal in the Dock to keep it handy.

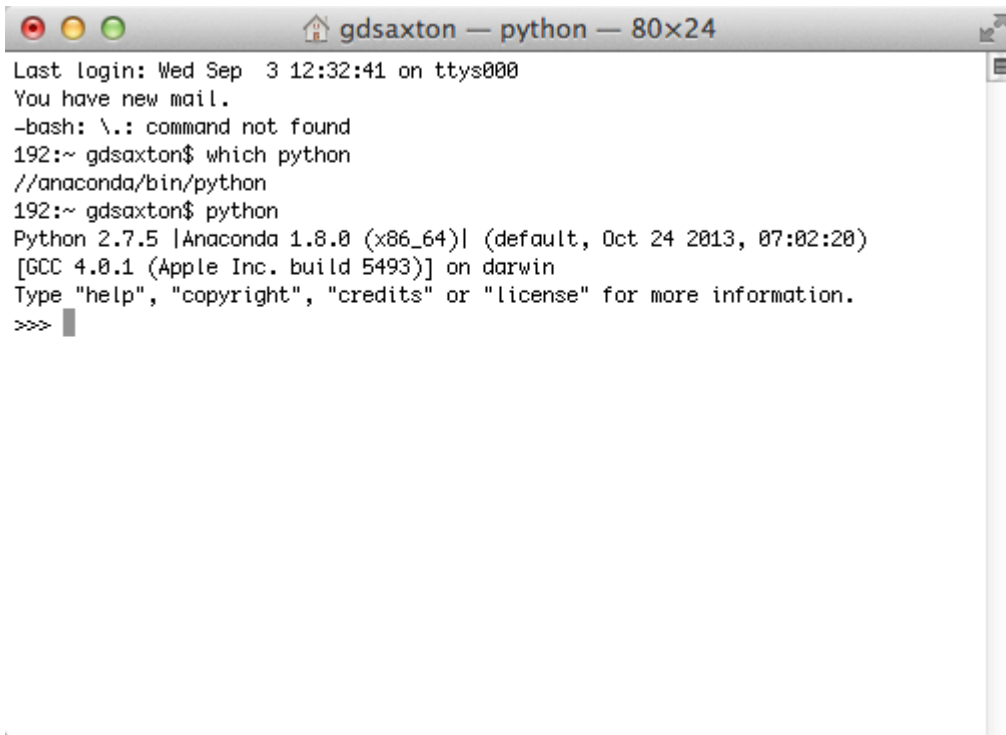
Open `/Applications/Utilities/Terminal.app` and find out which version of Python is the default by typing in `which python` at the `$` command prompt. It should look like the screenshot you see below. What you should get back is some indication that Anaconda Python is the default Python installation on your computer, such as `//anaconda/bin/python`

A screenshot of a macOS Terminal window. The title bar shows the user 'gdsaxton' and the shell 'bash' with a window size of '80x24'. The terminal output shows the last login time, a notification about new mail, and the execution of the 'which python' command, which returns '//anaconda/bin/python'.

```
gdsaxton — bash — 80x24
Last login: Wed Sep  3 12:32:41 on ttys000
You have new mail.
-bash: \.: command not found
192:~ gdsaxton$ which python
//anaconda/bin/python
192:~ gdsaxton$
```

## Step Three: Verify Python is working

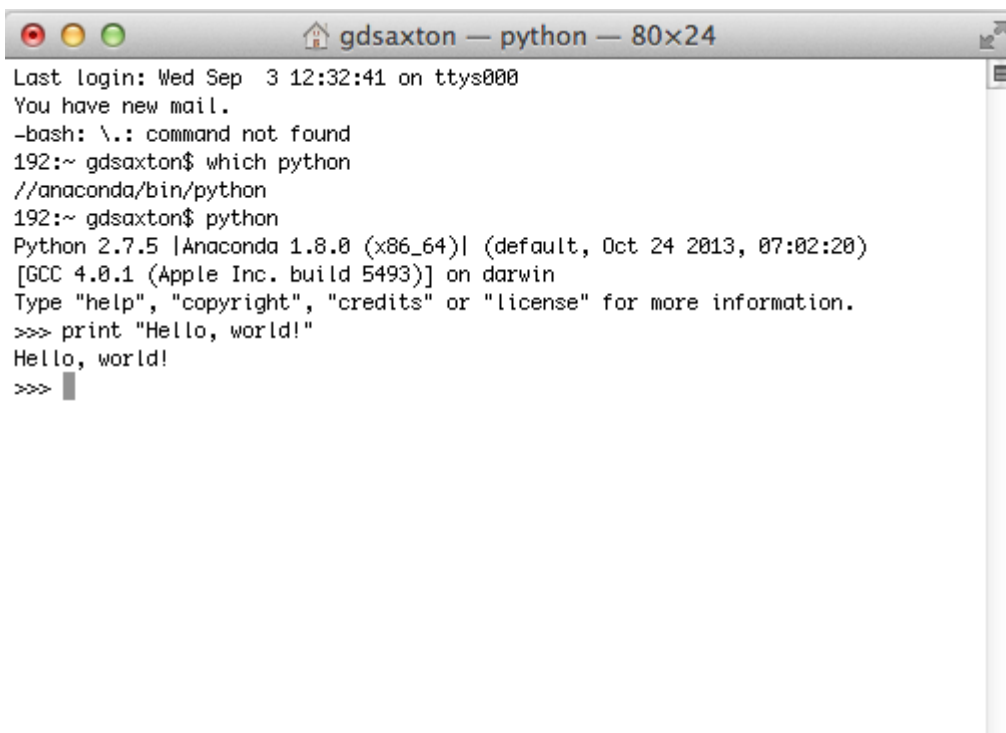
Start Python by typing in `python` at the `$` command prompt. You should see some language letting you know which version of Python has started and then a `>>>` command prompt. This is the prompt for you to enter Python code. Below I've included a screenshot of what this looks like on my device.



```
gdsaxton — python — 80x24
Last login: Wed Sep  3 12:32:41 on ttys000
You have new mail.
-bash: \.: command not found
192:~ gdsaxton$ which python
//anaconda/bin/python
192:~ gdsaxton$ python
Python 2.7.5 |Anaconda 1.8.0 (x86_64)| (default, Oct 24 2013, 07:02:20)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## Step Four: Type in Some Code

Let's type in the typical first command of a programmer, print "Hello, world!" and hit enter. The expected output, "Hello world", is seen in the following screenshot.



```
gdsaxton — python — 80x24
Last login: Wed Sep  3 12:32:41 on ttys000
You have new mail.
-bash: \.: command not found
192:~ gdsaxton$ which python
//anaconda/bin/python
192:~ gdsaxton$ python
Python 2.7.5 |Anaconda 1.8.0 (x86_64)| (default, Oct 24 2013, 07:02:20)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hello, world!"
Hello, world!
>>>
```

You've just executed your first Python command!  
Note that this is just one way of running your code. Specifically, when you're typing in commands in the Terminal you are working in the *interactive* mode. Your code is not saved, and thus the interactive mode is used only for simple tests. In the next tutorial I will walk you through two more powerful ways of creating and running Python code.

If you have found this helpful please share it, and I always welcome feedback.