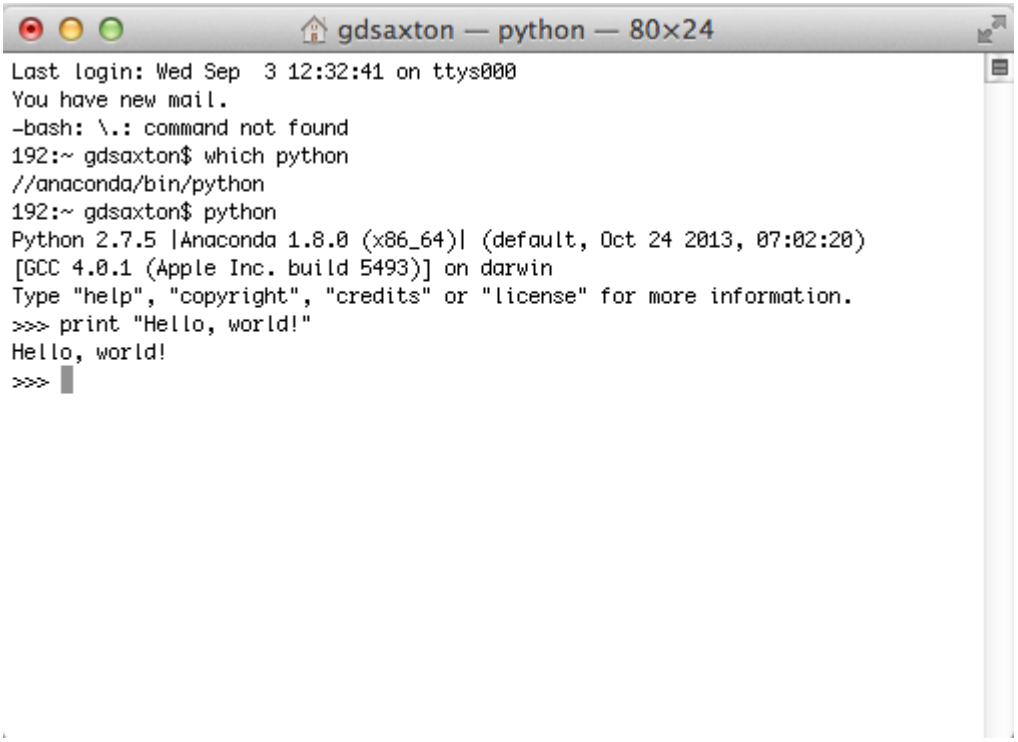
Your First Steps with Python: Part II – Four Ways to Run your Code

This is the second in a series of posts to get you up and running on Python. In the first post I showed you which version of Python to install, how to check that the installation succeeded, and how to type in and run your first simple Python command.

In this tutorial I will show you four different ways of writing and running your code. For simplicity's sake, each of these four methods will run the same typical beginners' "Hello, world!" code. For the purposes of the tutorial I am assuming you are using a Mac. Instructions will vary slightly for PC or Linux.

Method One: Interactive Mode

The most basic way to run code is to enter and run lines of code in the Terminal. This was covered in Part I of the tutorial series. To recap, open */Applications/Utilities/Terminal.app*, start Python by typing in python at the $ command prompt, type in print "Hello, world!" and hit enter. The expected output, "Hello world", is seen in the below screenshot.

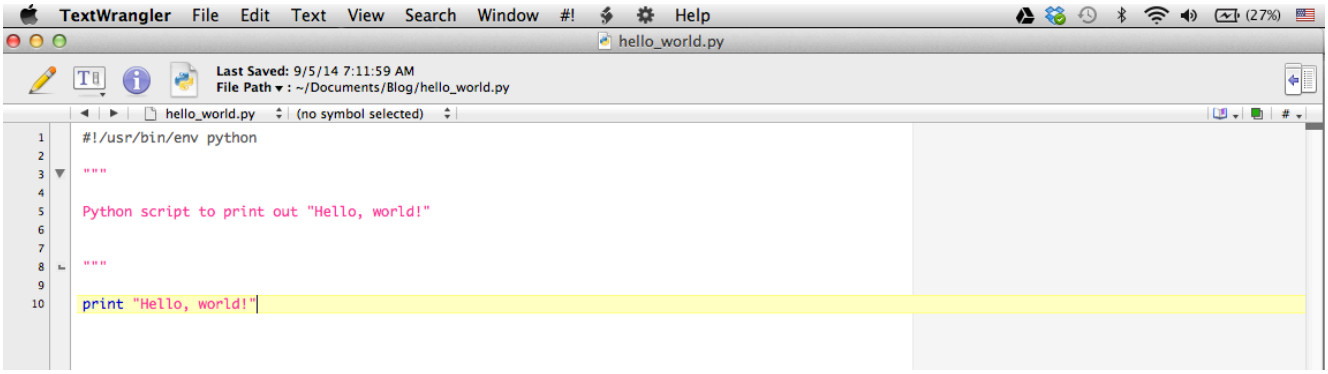


This *interactive* mode works, but is really only helpful for basic coding. Your code isn't saved, so you'll want want to take a different approach.

Method 2: Running Python from a Code-Friendly Text Interpreter

Python code is written in plain text files, typically given a ".py" extension. Once a script (e.g., *twitter.py*) has been written, it can then be "run" by Python and perform whichever tasks are laid out in the script. So, you'll need to find a good program for editing these text files. On my Mac I use TextWrangler. Vim is popular on other machines. Your choice here is not terribly important. These programs are all generally free and easy to install. One benefit you'll get from these programs over say, the basic TextEdit app on the Mac is that they will highlight various elements of the Python code syntax, which helps in code formatting. Running the code is also easier through these specialized programs. Let's say you download and install TextWrangler and enter your code.

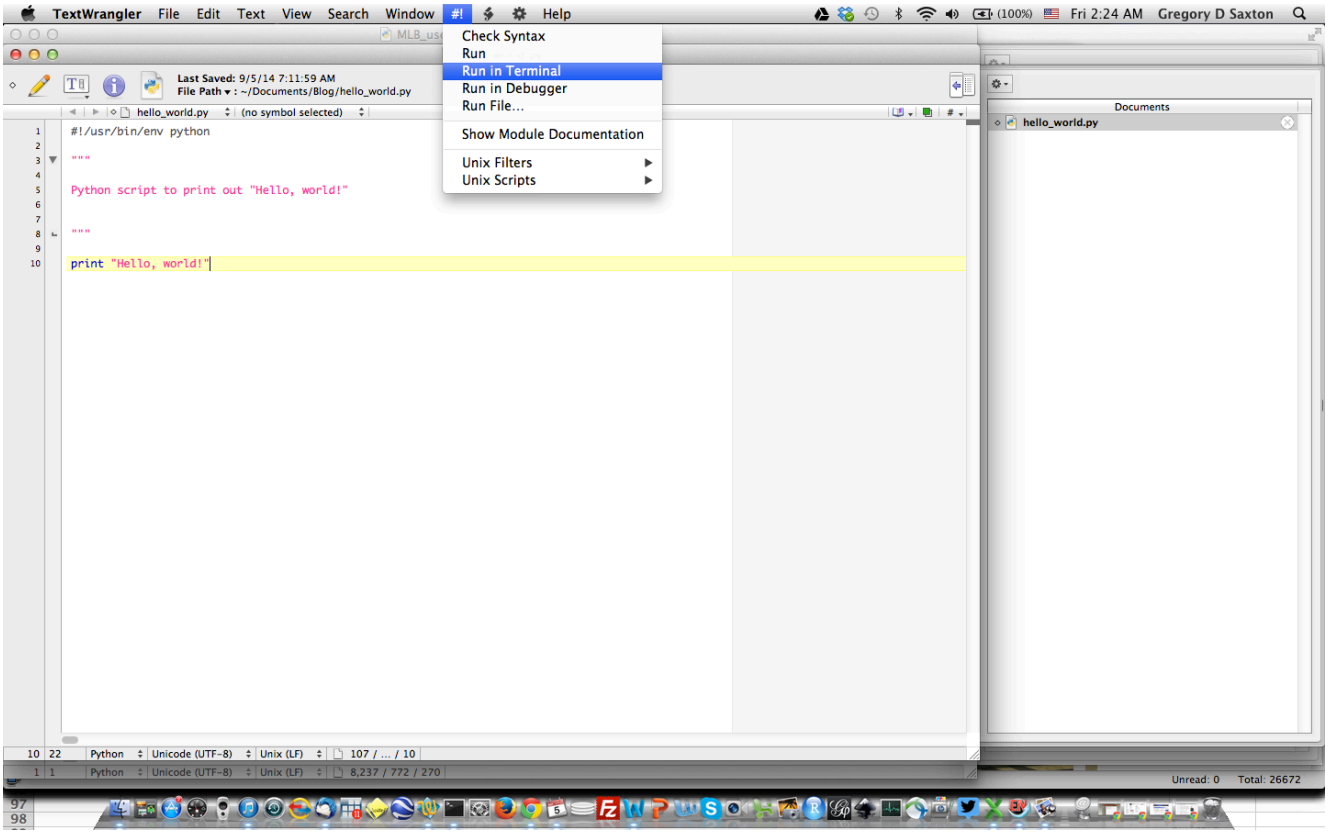
This is what is might look like.



The first line in the code is the shebang – you'll find this in all your Python scripts.
Lines 3 – 8 contain the docstring – also a Python convention. This is a multi-line comment demarcated by the docstrings """ that describes the code. Write whatever is helpful to you as well as anyone who might use your script in the future. For single-line comments, use the # symbol at the start of the line.

Line 10 contains the python code

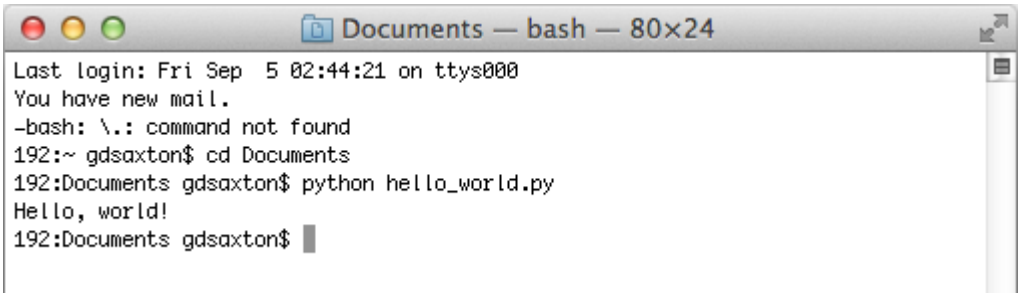
OK, so you have written and saved your code in a file called hello_world.py. You can now run your code through TextWrangler and other text interpreters. Go to the #! menu and select "Run in Terminal" as in the following screenshot.



What happens is that the Terminal will open another window and run your code. Below you'll see the output on my MacBook Pro.

Method 3: Running a Python Script in the Terminal

You don't want to do it this way. If you want to run your scripts, use Method 2 instead. But I'll show you quickly just so you know it's possible. Open up a Terminal window. Let's say you saved your script with the name hello_world.py in your Documents folder. Navigate to the Documents folder by typing in cd Documents and hit enter. Then to run your code type in python hello_world.py and hit enter. Your code will run and you'll see the output as shown below.

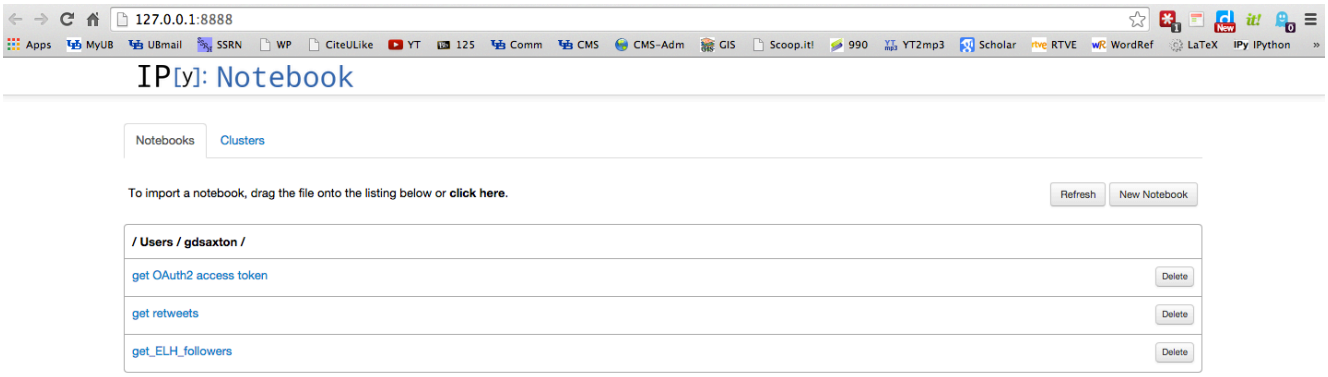


Method 4: iPython Notebook

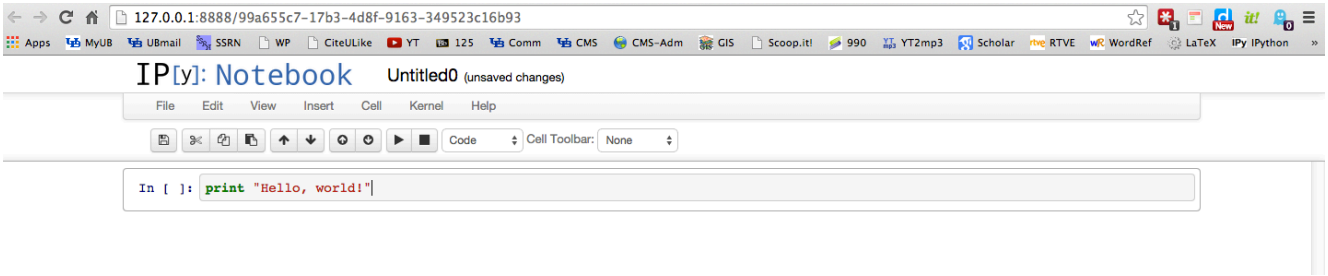
This is the preferred way for running your code. I recommend that you familiarize yourself with the iPython Notebook, which comes included with Anaconda Python. The link provides an overview of the Notebook. Simply put, it has become a boon for *interactive* code development, that is, for "playing around" with the code. I now use the iPython Notebook for developing all of my code. In the same window it allows you to write blocks of code, run them, check whether they worked as intended and, if not, modify them. Highly recommended for learning as it allows for quick error checking. Annotation of your code is also facilitated.

Here is what you'll do. Open up the Terminal and type in ipython notebook.

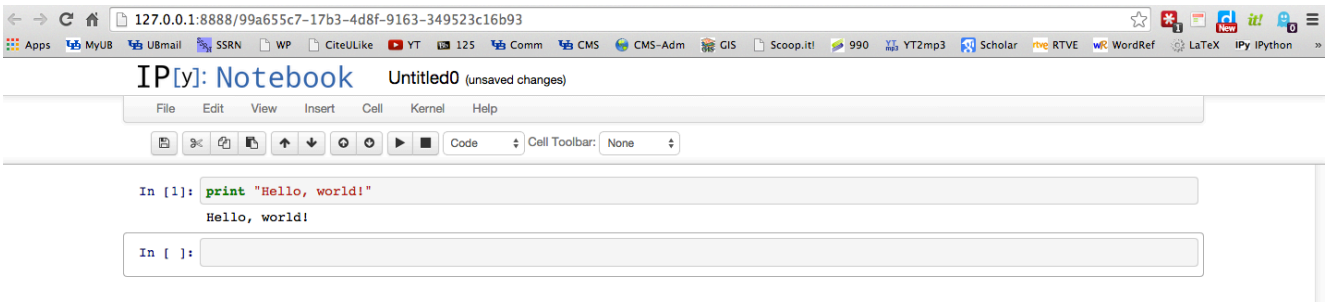
You'll see some lines of script running in your Terminal window indicating that the Notebook app is running. Wait a few seconds, and a browser window will open. This is your iPython Notebook interface. It will show all available *notebooks*, which are just fancy Python scripts you've developed in iPython. An example is below.



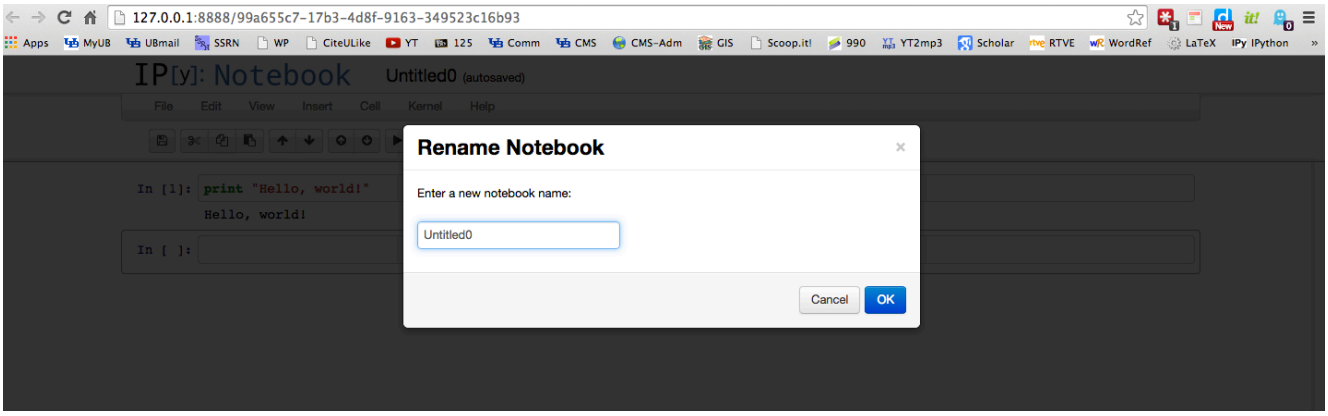
Hit *New Notebook* and another browser window will open. This is where you will type in your print "Hello, world!" code in the first cell, as shown below.



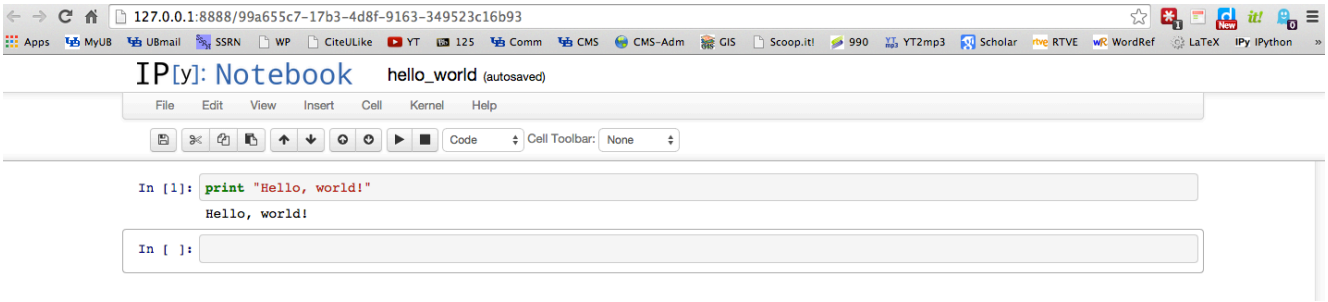
In iPython you can run each block of code, or cell, individually. Put your cursor anywhere in the first cell and hit Shift enter. You code will run. See below.



One final step remains. You'll want to rename and save your new notebook for future use. Click on 'Untitled0' and a dialogue box will open like you see below.



Type in hello_world and hit OK and you'll see that the code has been saved.



Et voila! You now know 4 different ways of running Python code. In general, you won't want to do methods 1 (interactive mode in the Terminal) or 3 (run scripts through Terminal). You may wish to use Method 2 from time to time by typing up entire scripts in TextWrangler, but that's a topic for another day. Until you learn a bit more, concentrate on the last method and do your coding in the iPython Notebook.

Happy coding!