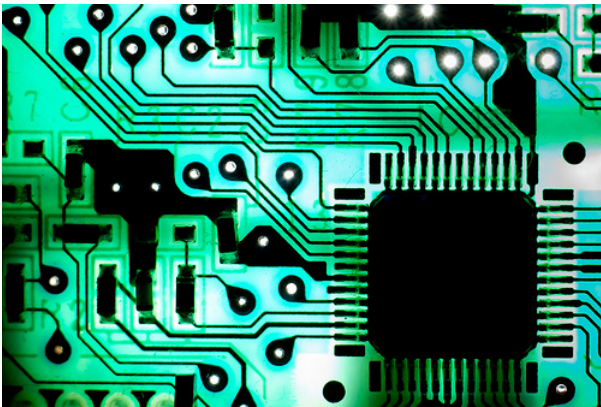


Python: Where to Start?

For the complete beginner, getting up and running in a new programming language can be a daunting task. I'll try to simplify it here by walking you through the key steps.

Choose a version of Python

There is more than one version of the Python language. Most people choose between version 2.7 and 3.3. Go for 2.7, because some of the utilities you'll be using won't be readily available in the newer 3.3.



Now you have to choose where to get the software. There are lots of choices. If you own a Mac, it comes with Python pre-installed. Forget about that one. For Unix, Windows, and Mac users alike I'd 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.

Familiarize Yourself with the Basics

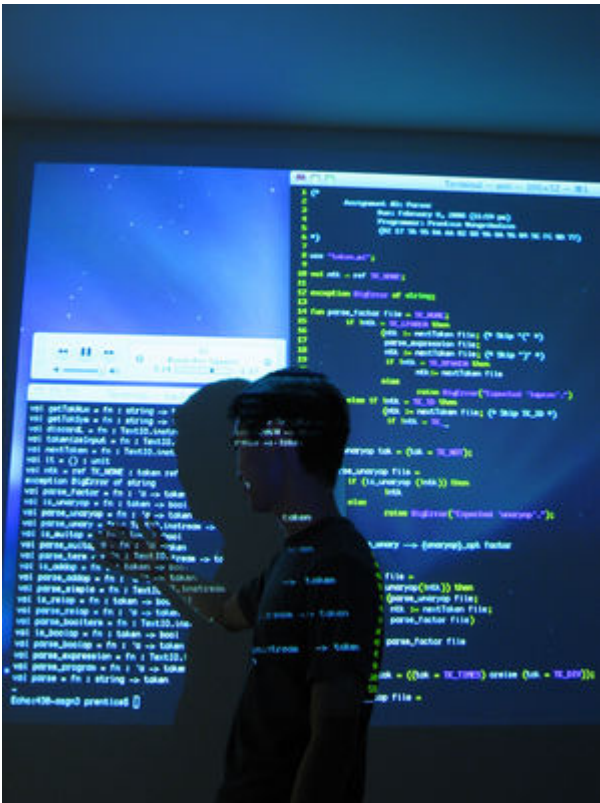
I recommend a two-pronged learning approach: 1) familiarizing yourself with the basics of the language and 2) playing around with other people's "recipes." For the former, I highly recommend doing some of the many excellent tutorials now available online for learning how to use and run Python. A

great place to start is [Codecademy](#). It takes an estimated 13 hours to complete this self-paced, 12-part interactive tutorial. Do one each day and you'll have a basic understanding in less than two weeks.

Play around with code – Step 1

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. The 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, as in the following example, which helps in code formatting:

I would also 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.



Play around with code – Step 2

This brings us to the second part of your learning strategy: playing around with other people's code. There are lots of places to find pre-existing code. Just Google it. Just start with something simple, copy it into the iPython Notebook or Textwrangler, etc., and run it. Modify. Play around with. Try to understand it. Annotate your code to help facilitate the learning process.

If you're interested in the same types of questions as I am you may find some of my code useful. A good place to start, once you've made it through some of the CodeAcademy tutorials, is my [Python Code Tutorial: Part I](#).

Ask Questions

You will run into errors. You will run into problems. You will run into roadblocks. The Web will be your friend. If you've checked your code, Googled error messages, and can't find an answer, [StackOverflow](#) will be your friend. It is likely

someone has already asked the same question there. If not, ask away and you will typically get an answer the same day. Just respect the community norms and you will find StackOverflow to be a tremendously helpful resource.

Contribute

Python and other open-source communities only thrive through the volunteer efforts of countless individuals across the globe. They will have contributed to your learning, so once you've developed sufficient knowledge give back. Answer others' questions on StackOverflow. Write a tutorial on your blog. Share your code on [Github](#).

Step-by-Step Tutorial

I've also just begun a series of [step-by-step tutorials](#) to walk you through installing Python and running your first code. Take a look if you're interested, and please share this on social media if you know others who may be interested.