

Here are a few useful Python-related resources (contributed by Charlie Sharpsteen).

The main documentation is hosted at <http://docs.python.org> and is very, very well-written and maintained. When I first started with the programming language, I worked through parts of their tutorial:

<http://docs.python.org/tutorial/index.html>

The tutorial contains a good amount of humor and plenty of references to Monty Python which keeps things interesting. Another introduction to the language that I have heard good things about is Learn Python the Hard Way:

<http://www.learnpythonthehardway.org>

After getting started with the language, be sure to keep a handy bookmark to the Standard Library reference:

<http://docs.python.org/library/index.html>

I usually have that page open in a browser when writing Python so that it is easy to look things up.

When using Python for scientific computing, one usually ends up calling on many libraries that use compiled code for efficiency. These libraries can be difficult to install on platforms like Windows. The Enthought Python Distribution is freely available to academic users and provides just about everything needed for scientific computing:

<http://www.enthought.com>

Another good choice is the PythonXY distribution which is freely available to non-academic users:

<http://code.google.com/p/pythonxy>

Enthought is also the company behind two libraries called NumPy and SciPy which provide efficient matrix operations and basic MATLAB-like functionality and therefore form the backbone of nearly all scientific Python code:

<http://scipy.org>

Finally, here are some tips I wish I had been aware of when I was just starting out with Python:

- Stick with the newest version of Python 2 and ignore Python 3 for now. Python 3 was a significant re-write language that was released in 2008. However, Python 3 is not entirely compatible with Python 2---the version of the language in which the majority of code is written. This means that a lot of useful libraries, especially scientific libraries, will not work with Python 3. If you write code that runs on Python 2.7.2, it will be easy to move to Python 3 once enough of the old libraries complete the migration that Python 3 becomes the version of the language most people use by default.
- Python can be used in an interactive manner, which makes it very useful for exploratory data analysis. However, the standard Python shell is a bit frustrating to use in this manner. As an alternative, use IPython (<http://ipython.org>) instead. This shell has indispensable features like tab-completion and easy

access to help and documentation along with a whole pile of other goodies.

- Once you start downloading and installing Python code written by other users, learn how to use the Pip package manager (<http://www.pip-installer.org>). The default tool for finding and installing Python code is called `easy_install` and it is absolutely horrible. Pip is much more user friendly. Chapter 46 of Learn Python the Hard Way covers this important tool: <http://learnpythonthehardway.org/book/ex46.html>