

Getting Started with Python

EE 106A/206A / BioE 125

UC Berkeley

For this course, you are expected to have familiarity with programming in general, though Python may not be as familiar to you as other languages. Most basic question you have about Python can be answered by looking at the official documentaion (available here: <https://docs.python.org/2/>). More advanced or specialized questions will usually be found on StackOverflow (<http://stackoverflow.com/>). Often, if you perform a Google search for something along the lines “python [problem]” or “how to [do something] in python,” a StackOverflow thread will be one of the top results.

For a tutorial that can help introduce you to the syntax and peculiarities of Python, there is always the official tutorial (here: <https://docs.python.org/2/tutorial/index.html>). Another option is the website <http://www.learnpython.org/>, which features and interactive Python interpreter to help you test code as you learn. Other options are listed at <http://www.codeconquest.com/blog/the-50-best-websites-to-learn-python/>.

If you are interested in using Python similarly to MATLAB (for interacting with and plotting data, manipulating matrices, etc.), the NumPy and SciPy packages are excellent options. They are open-source and free (unlike MATLAB), and you can find out more about them here: <https://www.scipy.org/>. You can also find a useful “translation” page from MATLAB to NumPy/SciPy here: http://scipy.github.io/old-wiki/pages/NumPy_for_Matlab_Users.html, and an overall description of MATLAB-Python similarities and differences here: <http://bastibe.de/2013-01-20-a-python-primer-for-matlab-users.html>.

For experimenting with Python on your own, or using it interactively like you would use the MATLAB command window, the iPython shell is an excellent option (available here: <https://ipython.org/>). It is similar to Python’s built-in interpreter, but offers greatly expanded functionality and improved ease of use.