Syllabus: Introduction to Python

Chang Y. Chung

January 2014

Figure 1: Python logo from `http://www.python.org`. The name is not after those dangerous reptiles; it is from the seventies comedy series "Monte Python's Flying Circus".

Introduction

Python is a popular, general-purpose, multi-paradigm, open-source, scripting language. It is designed to emphasize code readability – has a clean syntax with high level data types. It is suited for interactive work and quick prototyping, while being powerful enough to write large applications in.

Python has a large number of available and well-written modules for everything from abstract syntax trees to ZIP file manipulation. Its echosystem features extesive set of tools from JIT compiler[1] to fancy IDE[2]. In this half-day workshop, you will be introduced to basic Python language syntax and to its echosystem.

[1] PyPy (`http://pypy.org`)

[2] IPython (`http://ipython.org`)

Figure 2: IPython (`http://ipython.org`) is a rich architecture for interactive computing. Version 1.0.0 was released on Aug, 2013.

Objectives

After taking this course, you should be able to:

- use Python interactively
- execute a Python script at the shell
- use Python types, expressions, and None
- use string literals and string type (raw and unicode strings as well?)
- use Python statements (if...elif..else, for, pass, return, continue, and so on)
- understand the difference between expressions and statements
- understand assignment semantics
- write and call a simple function (lexical scoping rule?)
- utilize high-level data types such as lists and dictionaries
- understand the difference between mutable and immutable objects
- write a simple class and access methods and attributes
- import and utilize a module (Beautiful Soup?)
- read from and write to a text file
- understand interpreter and compilers: CPython, PyPy, Cython
- see demonstration of IDE's: IDLE, IPython, IPython Notebook, hosted environments
- understand the role of package managers: easy_install, pip
- understand what NumPy does and what SciPy is (are?)
- learn about resources for learning Python[3]

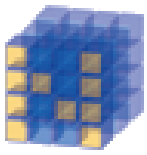

Figure 3: NumPy (`http://www.numpy.org`) implements an N-dimentional array and is considered as the fundamental package for scientific computing with Python.

[3] tutorials, books, MOOC's, videos, websites, Python Koans, Python Challenge, and Project Euler

Intended Audience

This workshop is for those who have some experience in using at least one scripting language[4] but who do not know Python. It is assumed that you can edit a text file using your favorate editor, and be able to execute your script file on the command line of a shell.

Python comes pre-installed in Mac OS X and Linux. Since Python 3 is not backward compatible and not all the modules are upgraded into Python 3, we will use the latest version of Python 2 (2.7 as of writing this) in this workshop, which should also be the default for Mac OS X and Linux.

A nice instruction for installing Python on Windows is at The Hitchhiker's Guide to Python site[5].

Installation of pip is optional but encouraged since pip is the tool for installing and managing Python packages. PIP installation guide is at `http://www.pip-installer.org/en/latest/installing.html`.

[4] Stata, R, MATLAB, Perl, Ruby, emacs lisp, bash, or PowerShell, etc.

[5] `http://docs.python-guide.org/en/latest/starting/install/win/`

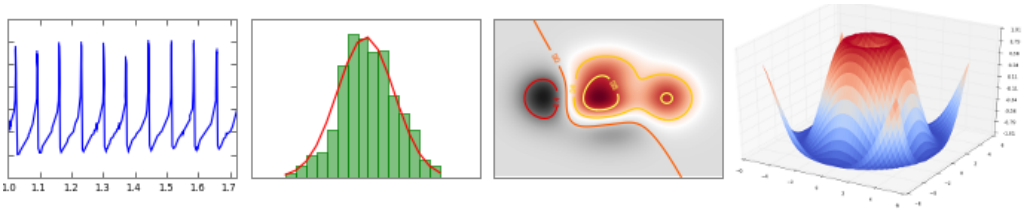

Figure 4: matplotlib module (`http://matplotlib.org`) is a python 2D plotting library by John Hunter, who unfortunately died from cancer in 2012.

Location

OPR Computer Lab: #217 Wallace Hall (or TBA)

Schedule

Section	Format	Topic
Basic Syntax	Lecture & Quiz	Hello World to List & Dict
Intermediate Syntax	Lecture & Quiz	mutable to file IO
Echosystem	Demo & QA	Compilers to Resources
(Optional)	Survey	Evaluation

Table 1: Each of the three sections will last 50 minutes total with a 15-20 minute break in between the section. At the end of each section, a homework or in-depth lab material will be provided.