Syllabus: Introduction to Python

Chang Y. Chung

May 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 ecosystem features an extensive set of tools including a JIT compiler[1] and fancy IDE's[2].

In this workshop, you will be introduced to basic Python language syntax and to its ecosystem.

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

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

Figure 2: IPython (`http://ipython.org`) is a rich architecture for interactive computing. Version 2.0 was released on April, 2014.

Objectives

After taking this course, you should be able to:

- use Python interactively
- execute a Python script at the shell prompt
- use Python types, expressions, and None
- use string literals and string type
- use Python statements (if...elif..else, for, pass, continue, . . .)
- understand the difference between expressions and statements
- understand assignment semantics
- write and call a simple function
- import and utilize a module
- read from and write to a text file
- utilize high level data structures: lists
- create dictionaries and complex data structures
- understand tuples and automatic unpacking
- understand the difference between mutable and immutable types
- write a simple class and access methods and attributes
- learn about how to handle and to raise exceptions
- learn about resources for learning Python[3]

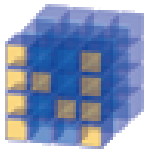

Figure 3: NumPy (`http://www.numpy.org`) implements an N-dimensional array and is considered as the fundamental package for scientific computing with Python.

[3] tutorials, books, MOOC's, videos, web sites, 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 favorite editor, and be able to execute your script file on the command line of a shell.

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

Environment

Python can be installed and used in different ways. In this workshop, I will use IPython Notebook environment for demonstration. All the lab PC's will have IPython Notebook installed.

If you rather use other environments, then you are welcome to bring your own laptop computer. Free Anaconda distribution[5] is recomended, but other environments (for example, command line Python and emacs editor) are OK.

[5] Available at: `https://store.continuum.io/cshop/anaconda/`

Since Python 3 is not backward compatible and not all the modules are upgraded into Python 3, we will use Python 2 (2.7 as of writing this), which is the default for IPython notebook in the Anaconda distribution.

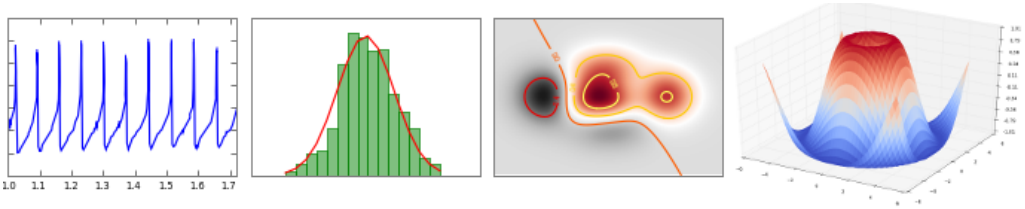

Figure 4: These plots are generated using the matplotlib module (`http://matplotlib.org`), which is a python 2D plotting library created by John Hunter, who unfortunately died of complications from cancer treatment in 2012.

Location and Date

Location: OPR Computer Lab: #217 Wallace Hall
Date: This workshop is offerred twice. Please signup for one either on Thursday (Thu., May 8. 9:30 am - noon) or on Friday (Fri., May 9. 9:30 am - noon).

Schedule

Section	Format	Topic
Basic Syntax	Lecture & Quiz	Hello World to File IO
Intermediate Syntax	Lecture & Quiz	List to Exception
(Optional)	Survey	Evaluation
(Self-study)	Ecosystem	Demo & Resources

Table 1: Schedule. We will spend most of the time going through the first two sections. For the third section (ecosystem and demonstration), a deck of slides (PDF format) and an IPython Notebook (for demonstration) will be provided for self-study after the workshop.