Python Intermediate Tutorial

Awn Duqoum

11th February 2017

1 Purpose

The purpose of this tutorial is to teach more advanced concepts in Python and structured programming in general. The purpose of this document is then to outline the structure of the tutorial and to provide the resources for further learning.

2 Required Software

In order to follow along with this tutorial, you will require the following:

- Python 2 (2.7.13)
  - Can be found at `https://www.python.org/downloads/`
- Notepad ++ (or any text editor you like)
  - Can be found at `https://notepad-plus-plus.org/download/v7.2.2.html`
- A sense of wonder
- A desire to learn

3 Forward

Even though this is a more in depth tutorial than the first one that was held, a basic understanding of python is helpful but not mandatory. Knowledge of other programming languages should make you capable of keeping up during this session. If you are worried the material will go over your head and would like to get a taste of python before starting, check out the beginner tutorial material here and feel free to ask a lot of questions during.

4 Outline

The code that will be written in each section can be found here. It is highly encouraged that you write the code yourself, mainly so you can get into the habit of writing it. In addition, you can leave comments for yourself so that you can use your comments from this tutorial for future reference. If, however, you cannot keep up to a section, DON'T WORRY—you can download the files, which are commented and easy to follow.

5 Programming Modules

1. Multidimensional lists
   - What are they?
   - Limitations - Don't get carried away with levels
   - Example : A Robot Factory
2. OS/Sys interfacing
   - Why does one need to interface with the OS ?
   - Examples of a few commonly used functions
3. Classes
   - What are classes ?
   - Running a class as a script
   - Creating a new instance of a class
   - Example : Directory Scrapper
4. Command Line arguments
   - Using vanilla (C-like) parsing
   - Using a more elegant approach - ”argpase” library
   - Example : Automating our class

6 Whats next?

Now that you have a intermediate understanding of Python, with this knowledge there is no limitations on what you can make. Below is a list of topics that extend these concepts that are worth looking into to learn more about how powerful this language is, note they are picked because Python is great at implementing them.

- Neural Networks
  – Note: This is really easy to do in Python - Talk to me at the end of the tutorial if you want a rundown and how and a link to sample code
- Machine Learning
- Image Analysis (openCV is your best friend)