

Workshop: Software Writing Skills for Young Researchers (23.-25. September 2015)

% Document created by Malvika Sharan

Python for Novice

Introduction

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. *A byte of Python*

There are several Python tutorials available for beginners, therefore rather than creating a new tutorial we considered putting them together.

Useful resources:

- Web links:
 - [Software Carpentry: Python lessons for novice](#)
 - [Geeting Started](#)
 - [Python Beginners Guide](#)
 - [Execute Python](#)
 - [Use Python](#)
 - [Code sample and snippets for Beginners](#)
 - [Python programmers Guide](#)
 - [MIT lecture notes](#)
 - [MIT: Introduction computer Science](#)
 - [Basic snippets](#)
- Interactive learning:
 - [Codecademy](#)
 - [After Hours Programming](#)
 - [More suggestions on \[Quora\]\(https://www.quora.com\)](https://www.quora.com)
- Books:
 - [A byte of Python](#)
 - [Learn Python the hard way](#)
 - [Dive into Python](#)
- Video tutorials:
 - [YouTube DRAPS TV](#)

- [Learn Python in one video](#)
- Many more... (literally millions of [YouTube videos](#))

Topics to be discussed

1. How To - 'Hello World'
2. Literal Constants: integers, floats and strings
3. Variables: use them in python commands (slicing, splicing, overwriting)
4. Operators and Expressions: plus, minus, multiply, power, divide, less/greater than, boolean and, or, not
5. Sets and Lists: define and access list (min, max, index, sort, unique, append, reverse, combine, sum, intersect, string to list etc.)
6. The if-statements: making choices and defining conditionals
7. For-loop: repeated tasks like reading multiple files for same action
8. Choosing editors to write Python programs
9. Basic rules: comments, quote, indentation, newline, tabs
10. File handling: create, read, write, open, close
11. Functions
12. Import packages and work with them: os, sys, numpy, matplotlib
13. Perform operations on arrays of data.
14. Display simple graphs
15. Basic error and exceptions: try and except

Editor that we will use in this course:

- [gedit](#) (installed)
- [PyCharm](#) (Alternative, not installed)

Debug:

<http://swcarpentry.github.io/python-novice-inflammation/09-debugging.html>