Outline: Python Basics

Seminar on Selected Tools Week 0 — Python, LaTeX and Git

ppppass

January 22, 2018

Section 1

Basic topics

Key properties of Python

1. Indentation based
2. Very-high-level
3. Interpreted: interactive, dynamic but slow
4. Object-oriented: "Everything is an object"
5. Partial functional programming support
6. Flexible and versatile: plenty of syntactic sugar
7. Glue language: easy to communicate with programs in other language
8. Easy to hook built-in utilities
9. Powerful libraries and strong community

Data structures

1. Numbers
2. Strings and bytes
3. Tuples
4. List
5. Dictionary
6. Set

Basic operations

1. Methods as attributes and objects
2. Pack and unpack
3. Keyword `del`: remove reference
4. Iterable as sequence

Flow control statements

1. Clause `if-elif-else`
2. Clause (`for`, `in`)-`else`: range-based loop
3. Clause `break`, `continue`, `pass`
4. Clause `with-as`: automatic clean up

Exception handling

1. Clause `try-except-else-finally`
2. Keyword `raise`
3. Class `Exception`: inheritance-based exception handling

Functions

1. Keyword `def`
2. Positional arguments and keyword arguments
3. Arbitrary argument lists: dynamic argument processing
4. Documentation strings: in-line helps and `help()`

Section 2

Further topics

Section 2

Further topics

Classes

1. Keyword `class`
2. Method `__init__()`
3. Class objects
4. Instance objects
5. Method objects
6. Class variables and instance variables

Modules

1. Keyword `import`, `from` and `as`
2. Attribute `__name__`
3. Function `dir()` and `help()`

Functional programming utilities

1. Lambda expressions
2. Iterables and iterators: lazy evaluation by `iter()` and `next()`, implemented by `__iter__()` and `__next__()`
3. Generator, `yield`
4. Comprehensions: list, dict, set and generator

Iterables manipulating functions

1. Function `items()`
2. Function `enumerate()`
3. Function `zip()`
4. Function `reversed()`
5. Function `sorted()`

Section 3

Miscellaneous topics

Input and Output

1. Method `format` of class `str`, or `printf`-style `%`
2. Function `open()` and `close()`
3. Method `read()` and `write()`

Dunder names

Dunder methods, or special methods, are a series of methods meant to be called by Python.

1. Module related: `__name__`, `__version__`
2. Type cast: `__str__`, `__int__`, `__float__`
3. Formatting: `__repr__`, `__format__`, `__bytes__`
4. Emulating callables: `__call__`
5. Emulating iterables: `__iter__`, `__next__`
6. Operators: `__le__`, `__lt__`, `__add__`

Section 4

Related resources

PEPs

1. Python Enhancement Proposals
2. PEP 8: Style Guide for Python Code
3. PEP 7: Style Guide for C Code
4. PEP 20: The Zen of Python
5. Try `import this` in interactive mode

Two key ideas of PEP 8

1. Readability counts
2. Consistency

Guidelines in PEP 8

1. Use 4 space per indentation level
2. Maximum line length of 79 characters
3. Blank lines
4. UTF-8 encoding
5. White spaces
6. Naming conventions

Related tool chains

1. Anaconda: a package and environment manager of Python
2. Package `venv`: a virtual environment manager
3. Program `pip` and PyPI: python package manager
4. Jupyter notebook: Web interactive Python application, especially live code demonstration and visualization