Practical Python

A Python language tutorial,
concepts and syntaxe in a practical
and direct approach.

Celso Providelo <cprov@canonical.com>

Canonical Ltd.

What is Python

- Interpreted Language
- Clarity and Simplicity
- Blocks controlled by indentation
- Dynamic Typing
- Advanced Types
- Objected Oriented

Why Python

- Simple concepts
- Simple syntaxe, clear and expressive
- Powefull pre-defined types
- Interactive interpreter
- Several libraries
- C e C++ extensions
- Multi-platform
- Free Software

Basic Python: Interpreter

- Using the interactive interpreter
  - 2 + 2
  - *print*
  - Escape: *CTRL-D*
- Creating and executing code
  - *Unix-like: #!/usr/bin/env python*
  - *Win32: HelloWorld.py*

Basic Python: Types, variables and values

- Numerics: a = 2
  - *int, float, bool, complex*
- Lists: L = [1, 2, 3, 5, 8]
- Strings: S = “Hello World”
- Tuples: T = (“Romeo”, “Juliet”)
- Dictionaries:
  D = {“name”: “Karl”, “age”: 28}
- Set: set([1, 2, 3])
- *type(var)*

Basic Python: Operators

- Aritimetics: +, -, /, *, %, **
- Bit-wise: &, |, <<, >>, ~
- Attribution: =, +=, -=, /=, *=
- Conditional: ==, !=, <, >, <=, >=, in
- Logics: and, or, not, is
- Combining operators
- Replacing in strings

Basic Python: Control Structures

- Conditional: if a == b
- Interactive loop:
  - for l in L, for s in S, for k in D.keys()
  - break & continue
  - else
- Condictional loop:
  - while i < 5
  - break
  - else

Basic Python: Functions

- Syntax: def foo(args)
  - def excuse(reason):
- Default value
  - def excuse(reason=”sick”):
- Arguments manipulation: *args, **kw
  - def excuse(*args):
  - def excuse(**kw):
- Local & Global scope

Basic Python: Standards - I

- range(a, b)
- len(L)
- round(f, p)
- chr(i) / ord(c) / unichr(i)
- min(a, b) / max(a, b) / abs(n)
- hex(n) / oct(n) / float(n) / int(f)
- list(args) / str(args)

Basic Python: Standards - II

- open()
  - read()
  - readline()
  - readlines()
  - write()
  - seek()
  - close()
- raw_input()

Basic Python:Exceptions

- Concepts
  - What is an Exception
  - How to raise an Exception
  - tracebacks
- Manipulating Exceptions:
  - try/except
  - else

Basic Python: *Import* and Docstrings

- *import os*
- PYTHONPATH
- *dir() & reload()*
  - *dir(os)*
- *help() & docstring*
  - *help(os)*
- Documentation Tools: *epydoc*

Object Orientation (OO)

- Concepts
  - OO vs. Procedural Programming
  - Domain / Problem
- Objects, Classes e Instances
- Inheritance
  - Multiple Inheritance and getattr()
  - Class and Instance attributes
- Introspection, Reflection e Metadada

Relevant Modules

- *sys*
- *time*
- *os*
- *string*
- *math*
- *optparser*
- *urllib*
- PIL
- OpenGL
- NumPy
- pyGTK
- DB-API
- Zope
- Twisted

Thanks !

Christian Reis,

Prof. Cláudio Kirner

UNASP

Canonical Ltd