Python Mini Tutorial

Shui Hung Kwok
W. M. Keck Observatory
September 7, 2006

What is Python?

- Programming language
  - Created by Guido van Rossum
- Interprets compiled byte code
- Dynamically typed
  - Type goes with values instead of containers
- First released in 1991
  - Smalltalk:71, Perl:1987, Tcl:1988, Java:1990, PHP:1994
- Current version: 2.4.3 (March 2006)

Download and Installation

- http:/ /www.python.org/download
- See GettingPythonSOAP.html
  - fpconst and SOAPPy
  - MySQLDB
  - Python Imaging Library
- Start python

  $ python

Hello World!

```
Python 2.4.3 (#2, Jun 17 2006, 22:02:40)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> print “Hello World!”
Hello World!
>>> print 2+3
5
>>> Crtl-D to quit
```

Hello World!

```
print “Hello World!”
```

- Save in file ex1.py

- $ python ex1.py
- Hello World!

Interesting Data Types

- Sequences
  - Lists
    - a = [1,2,3]
  - Tuples
    - b = (1,2,3)
    - s = “This is a string”
- Dictionaries
  - ie {'key': 'value', 'a' : 'b'}, d['k'] = 'v‘
- Sets

mutable

inmutable

Statements

- Assigments
  - `a = 1`
  - `a,b,c = 99, 41, 5`
  - `a,b,c = s`
- Special statements
  - pass
  - global
  - print
  - del, raise, assert, exec

Control Flow Statements

- If ... [elif ... else]

```
if a == c:
    print "a equals c"
else:
    print "a and c are different"
```

- While ... [else]

```
while a == b:
    work ()
else:
    finalize ()
```

For Loops

- For loop

```
for item in sequence:
    Statements
else:
    statements
```

  - Example:

```
for item in (1,3,5,7,9,15,2003):
    print item
for item in xrange (1, 50):
    sum += item
```

Exceptions

- assert expr, [message]
- try ... except ... else
- try ... finally
- raise ...

Functions/Methods

- def ***functionName*** (params…):
- Parameters

```
def func1 (a,b,c):
def func2 (a,b,*t): # t is a tuple
def func3(u,*v,**w): # v is a tuple, w is a dictionary
```

- Invocation

```
func1 (1,2,3)
func1 (c=5, b=2, a=2)
func2 (1, 2, 99, 98, 95) # t=(99,98,95)
func3 (1, 2, 99, h=9, k=8) # t=(99), w = {'h':9,'k':8}
```

Object-Oriented Python

- Class ClassName [(baseClass1 [, baseClass1]...)]:
- Example:

```
class DemoBase:
    def method1 (self, arg1):
        print "Demo Base", arg1

    def method (self):
        print "Demo Base method"

class Demo1 (DemoBase):
    def method1 (self, arg1):
        print "Demo2", arg1
```

Special Methods

- Constructor
  - `__init__ ()`
- Destructor
  - `__del__ ()`
- Operators
  - `__add__(), __mul__(), ...`
- Misc
  - `__repr__(), __str__(), ...`

Anatomy of Python Modules I

import statements

class definitions

main statements

Anatomy of Python Modules II

- Import statements
  - import module [, module] ...
  - from module import object [, object] ...
  - Example:
    - `import os, sys`
    - `from sys import path`
    - `from sys import *`

Others

- Generators
  - Methods that remember their internal state when invoke multiple times.
- List Comprehensions
  - ie [x for x in sequence if condition]
- Lambda operator
  - Anonymous function
  - ie lambda x : 3*x + 4

A short Python program

```
import sys
table = {}
for line in file (sys.argv[1]):
        for word in line.split ():
        try:
                table[word] += 1
        except:
                table[word] = 1
wc = 0
for word, cnt in table.items():
        print word, cnt
        wc += cnt
print "Total %d words" % wc
```

Program argument

Try first, fix later

Reading XML

```
""" readXML.py
"""
from VOTable import *
import sys

rxml = VOXML ()
rxml.parse (sys.argv[1])
try:
        print rxml.getContent (sys.argv[2])
except:
        print "not found"
```

Instantiate new object

Parse file

Get content of this node

SOAP with SOAPpy

```
class sesame:
    def __init__ (self):
        self.wsdl = 'http://cdsws.u-strasbg.fr/axis/services/Sesame?wsdl'
        self.config = SOAPConfig ()
        self.config.namespaceStyle = '2001'
        self.config.typed = 1
        self.config.buildWithNamespacePrefix = False
        self.proxy = WSDL.Proxy (self.wsdl, config = self.config, noroot = 1)
        self.config.argsOrdering = {'sesame': ('name', 'resultType', 'all',
            'service')}

    def resolve (self, name):
        res = self.proxy.sesame (name=name, resultType='xpi',
            all=Types.booleanType (1), service='SNVA')
        self.xml = VOTable.VOXML (StringIO (res))
        return self.xml.root.Sesame.Resolver
```

MySQL Interface

```
class spocsDB:
    def __init__ (self):
        """ Connects to mysql server and gets a cursor
        """
        self.db = MySQLdb.connect (host = 'localhost',
            user = ‘skynode', passwd = ‘nvo', db = 'spocs')
        if not self.db:
            print 'null db'
        self.cursor = self.db.cursor (MySQLdb.cursors.DictCursor)

    def query (self, qstr):
        self.cursor.execute (qstr);
        return self.cursor.fetchall ()
```

MySQL Example

…/python/tutorial

$ python spocs.py “select * from spocs limit 10”

{'Ni': 0.0, 'CRMS': 1.8, 'Teff': 5770, 'Na': 0.0, 'Metal': 0.0, 'NKeck': 6,
'NAA

T': 0, 'logg': 4.4400000000000004, 'Si': 0.0, 'Vrad':
-0.10000000000000001, 'Fe'

: 0.0, 'ra': None, 'Ti': 0.0, 'NLick': 0, 'LRMS': 1.0900000000000001,
'dec': Non

….

Resources

- Python Home page
  - http://www.python.org/
- Tutorial by Guido van Rossum
  - http://www.python.org/doc/current/tut/tut.html
- Quick Reference
  - http://rgruet.free.fr/PQR24/PQR2.4.html
- Pydoc – documentation server
  - pydoc –p portNr
  - http://localhost:portNr/