Introduction to Python Programming

"Overview and basic programming constructs"

Advanced Programming

Shakirullah Waseeb
shakir.waseeb@gmail.com

Nangarhar University

February 28, 2018

Agenda

Programming with Python

- *Guido van Rossum* created the Python programming language in late 1980s
- Python strives to provide simple but powerful syntax
- Used at Google, Yahoo, Facebook, and NASA etc.
- In late 2008, Python 3.0 was released, earlier version 2.0

Python program execution

- Using Interactive Interpreter
  - >>> 5+5
  - >>> 10
- Using a text editor and then interpret it using Python interpreter
  - edit your code in a file with extension **.py**
  - python *abc.py*
  - or use shebang (#! path to the python interpreter) notation on the first line (#! /usr/bin/env python)

A code sample

- print "great language"
- print('great language')
- x = 35

-
  ```
  if x == 35 and 1==1:
      print 'True'
  elseif x == 1:
      print 'One'
  else:
      print 'Nothing'
  ```

Agenda

Some operators

- = is for assignment, and == is for comparision
- +, -, *, /, % are for arithmetic
  - special use of + for string contention
  - special use of % for string formatting
- instead of symbols words (and, or, not) are used as logical operators
- print is basic printing command
- The first assignment to variable creates it

Agenda

Basic Data Types

- **Integers** (default for numbers)
  z = 7/3
- **Floats**
  z = 3.5
- **Strings**
  'xyz' "xyz"

Agenda

Sequence Types

- **Tuples** a simple ***immutable*** ordered sequence of items, items can be of mixed types, including collection types
  >>> tup = (10, 18, 'xyz', (3,5), 5.6, 'fet')
- **List** mutable ordered sequence of items of mixed types
  >>> lst = ['xyz' ,5, 2.5]
- **Strings** immutable, conceptually very much like a tuple, strings are defined using quotes(",',""")
  >>> str = "some's string"
  >>> str = 'some string'
  >>> str = """some multi-line
  string"""

Sequence Types –continue

- individual members of *tuple*, *list*, and or *string* can be accessed using bracket "array" notation

```
>>> tup = (10, 18, 'xyz', (3,5), 5.6, 'fet')
>>> tup[1] # Second item in the tuple
18
>>> lst = ['xyz' ,5, 2.5]
>>> lst [1] # Second item in the list
5
>>> str = "some's string"
>>> str[1] # Second character in string
'o'
```

White spaces

Whitespaces has meaning in Python; especially indentation and newline

- **newline** indicates a new statement
- use consistent indentation to mark blocks
  - first line with **less** indentation is outside of the block
  - the first line with **more** indentation starts a nested block
- Often a colon appears at the start of a new block (e.g. for loops, decisions, functions and class definitions.)

Functions

- **def** creates a function and assigns it a name
- arguments are passed by assignments
- **return** sends a result back to the caller
- arguments and return types are not declared

function definition syntax

```
def <name> (arg1, arg2,......, argN):
    <statements>
    return <value>


def add (a,b):
    return a+b;
add (5,6)
11
```

Classes and Objects

class and object example

```
class <name> (object):
    <member functions and variables>


class calculator (object):
  def __init__ (self,a,b):
      self.a=a
      self.b=b
  def add (self):
      return self.a + self.b
  def mul (self):
      return self.a * self.b


cal = calculator(4,9)
cal.add()
cal.mul()
```

Your Turn: Time to hear from you!

[1]

[1] https://fensafitters.files.wordpress.com/2013/07/3d095.jpg

References

Matt Huenerfauth, Guido van Rossum, Richard P. Muller
Introduction to Python
October 19, 2009
http://tdc-www.harvard.edu/Python.pdf

Richard L. Halterman
Fundamentals of Python Programming Draft.
Southern Adventist University, January 18, 2018.
http://python.cs.southern.edu/pythonbook/pythonbook.pdf

Robert Zuber
Python Fundamentals Training.
Markana, April 2013.
http://simeonfranklin.com/python_fundamentals.pdf