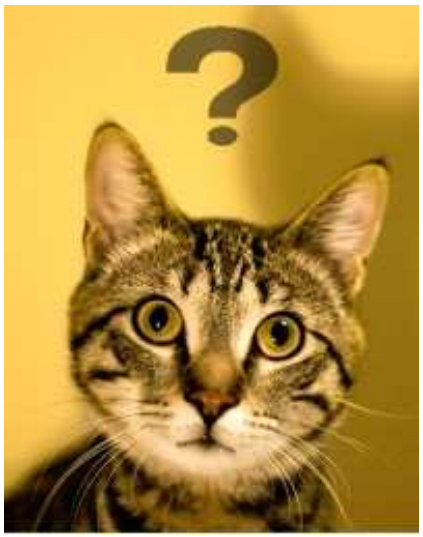
A Python Crash Course

Alex-P. Natsios – George Stoumpos

Technological Institute of Larissa

5 Dec 2012

A few words about python



Characteristics:

- created and published by Guido Van Rossum in 1991
- Interpreted
- Multiplatform
- Object Oriented
- Dynamic Typing
- Coding Style

How to Start Python?

You can either run python from a script or Interactively:

Interactive:

(using the interpreter to run python code)
$ python
>>> print "Hello there!"
Hello there

From a script:

(using a text file with python code)
$ python myscript.py
Hello there!
$

Data types in Python

Numbers: Integer(42), float(13.37), complex(12+3n)
Strings: "nothing interesting here move along"
Dictionaries: dict = {name : 'Maria', age : 22, 42 : 'haha' }
Lists: list = [1, 'classic', 2, 'boring']
Tuples: tuple = (1, 2, 42, 'Maria', (1,2,3))
Objects: instance = Class('foobar')
Modules: import antigravity

Python has objects!

Everything in python is an object and every object has its attributes.

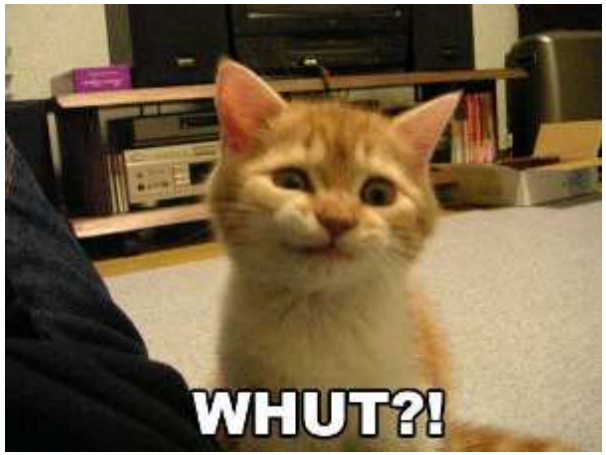
Object Attributes:

```
>>> name = 'maria'
>>> name.index('i')
3
```

breaking down the above statement we have:

- variable
- delimeter
- attribute
- parameters (if any)

Another peek in Attributes/Methods



Attributes?! Methods?! i can't remember all these stuff!

Lucky you! you dont have to know all of them

using dir(): dir allows you to peek into an object and learn about its supported methods
lets take a look at a list for example. >>> dir(list):
['append', 'extend', 'insert', 'remove', 'pop', 'index', 'count', 'sort', 'reverse']

Example

```
>>> dict = {name : 'Maria', age : 22, 42 : 'haha' }
>>> dict['age']
22
>>> 'name' in dict
True
>>> del dict['age']
```

(we could also use pop to also return the removed key's value)

Example

```
>>> list = [1, 'classic', 2, 'boring']
>>> list.append("Maria")
>>> list
[1, 'classic', 2 'boring', 'Maria']
>>> list.insert(4,"Alex")
>>> list
[1, 'classic', 2 'boring', 'Alex', 'Maria']
```

Tuples also known as Sequences and as such they are **ordered**.
you can also slice them using the **slice operator:**

Slice Operator

mysequence[i : j]
with 'i' and 'j' being your range in the sequence (i = start , j = one past the end)

Example

tuple = (1, 2, 42, 'Maria', (1,2,3))
>>> tuple[0]
1
>>> tuple[3]
'Maria'
>>> tuple[1:2]
(1,42)
>>> tuple[-1]
(1,2,3)
>>> tuple[3:]
('Maria',(1,2,3))

Python type mutability

The value of some objects can change. Objects whose value can change are said to be mutable;
objects whose value is unchangeable once they are created are called immutable.

mutable

Dictionaries, Lists

immutable

Numbers, Strings, Tuples, Frozensets

IF statement

```
name = "Maria"
if name == "Maria":
    print "Hello Maria!"
elif name == "George":
    print "Hello George!"
else:
    print "Hello stranger!"
```

While loop

```
i = 0
while i<5:
    print i
    i += 1
```

for loop

```
for item in range(1,100):
    print item
```

yet another for loop

```
names = ("Maria", "George", "Helen", "Alex")
for item in names:
    print item
```

Numeric Operators

+ - * /	Your everyday numeric operations.
%	Modulus - Divides left hand operant by right hand operant. Returns 'Remainder'
**	Exponent - performs exponential calculation on operant.
//	Floor Division - division operant where the digits after the result's decimal point are removed.

Function Definition

a simple function

```
>>> def foo(bar):
        print bar
>>> foo(123)
  123
```

Class Definition

a simple class

```
class AddressBookEntry(object):
    def __init__(self, name, phone):
        self.name = name
        self.phone = phone
    def update_phone(self, phone):
        self.phone = phone
```