

An introduction to the Python Programming Language

Pradeep Kishore G.
MindScape Computing
Bangalore

Who am I?

- PESIT ~ 1996–2000
- Programming since 13
- GNU/Linux user since '97
- Python Programmer since last 3 years.
- Used Python for AI, machine learning and visualisation.
- GUI,COM apps in Python to develop desktop apps/servers for Linux and Windows
- Write regularly about FOSS/Python
- Speaker at Linux Bangalore '04
- I use python professionally (Zope Application Platform).

What is Python?

- Interpreted, Object-Oriented, high level programming Language
- “Features”
 - Simple, easy to learn, easy to “read”
 - Free and open source
 - Portable and interpreted
 - Object-oriented
 - Extensible
 - Embeddable
 - Extensive libraries

Why Python

- Programs should be written for people to read, and only incidentally for machines to execute. –*SICP*
- Productivity over Performance
- Python lets me concentrate on the “**concepts**”
- It fits in your head. “*There's only one way to do it; the right way*”
- Batteries Included
- Plays well with others ...

The Zen of Python

- **Beautiful is better than ugly.**
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- **Readability counts.**
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- **There should be one-- and preferably only one --obvious way to do it.**
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- **If the implementation is hard to explain, it's a bad idea.**
- **If the implementation is easy to explain, it may be a good idea.**

Who Uses Python?

- Sshhh... its our secret sauce
- Google, Yahoo!
- NASA,NSE
- RedHat (GNU/Linux), Apple(Mac OS X), Microsoft
- Bittorrent (carries more than 40% of all the internet traffic)
- More success stories at:
www.pythonology.com

Before getting started..

- *Everything* is an object (even functions!!)
- Indentation matters!
- Use the interpreter, luke!
- Learn to use the help & `__doc__`
- strict type checking
- Unlearn ~~bad~~ old habits

Control Flow

- **if** statement

```
if temp == 19:
```

```
    print 'ambient'
```

```
elif temp > 19 and temp < 24:
```

```
    print 'warm'
```

- **for** statement

```
for i in range(10):
```

```
    print i
```

```
.....
```

```
fruits = ['apple', 'banana', 'guava', 'orange']
```

```
for fruit in fruits:
```

```
    print fruit
```

Functions

Defining...

```
def sos(a=3,b=4):  
    '''sum of squares'''  
    sum = a*a + b*b  
    return sum
```

calling...

```
sos1 = sos() # returns 25  
sos2 = sos(4) # ?  
sos3 = sos(5,5) #?
```

- Named parameters
- Default Parameters
- Comments and Doc strings (!!)

Data Structures

- Lists – []
- are mutable....

```
cart = []
```

```
cart.append('mango')
```

```
cart[0] # returns mango
```

```
cereals = ['rice','wheat']
```

```
cart.append(cereals) #2D list
```

```
['mango',['rice','wheat']]
```

- `dir(cart)` #what more can i do with a list or <any object>?

- Tuples – ()
- are immutable ...

```
nation = 'India'
```

```
capital = 'New Delhi'
```

```
print 'Capital of %s is %s' % (nation, capital)
```

Data Structures...

- Dictionaries
- are just hashes

```
mailid ={'pradeep':'pradeep@btbytes.com'  
        'bangpypers':'bangpypers@yahoogroups.com' }  
print mailid['pradeep']  
print "there are %s items "% (len(mailid) )
```

- Splicing

```
a = range(1,10)  
print a[2:] #index 2 onwards  
print a[2:5] # 2 onwards, upto 5  
print a[:8] #first 8 elements  
print a[-2] # 2nd element from right
```

File Handling

name, age, email

Ram, 21, ram@ram.com

Sam, 22, sam@sam.com

Kabir, 26, kabir@kabir.com

to

students =

```
[['Ram', '21', 'ram@ram.com'], ['Sam', '22', 'sam@sam.com'], ['Kabir', '26', 'kabir@kabir.com']]
```

```
import string
```

```
data = file('students.csv', 'r')
```

```
students = []
```

```
for line in data.readlines():
```

```
    line = line.rstrip('\n')
```

```
    item = line.split(',')
```

```
    students.append(item)
```

Modules

- Modules are used to group similar functions and classes together
- There is no special syntax, just copy the functions into one file. and thats it
- using modules ...

```
from mymodule import sos
```

```
from mymodule import ShoppingCart
```

Classes

```
class Person:
    def __init__(self,name):
        """ the ctor"""
        self.name = name

    def greet(self,greeting):
        """ ___ """

        print 'Hello %s, %s!!' % (self.name, greeting)

luke = Person('luke')

luke.greet('May the source be with you')
```

Operator Overloading

- `__doc__`
- `__eq__` [=]
- `__ge__` [>=]
- `__le__` [<=]
- `__mul__` [*]
- `__add__` [+]
-

```
>>> import operator
>>> dir(operator)
for more details
```

Inheritance

```
>>> class Student(Person):
...     def __init__(self,name,regno):
...         Person.__init__(self,name)
...         self.regno = regno
...
>>> p = Student('pradeep','2005CS001')
>>> p.name
'pradeep'
```

• **All methods in Python are virtual**

Extending base class function



Multiple Inheritance


```
class C(A,B):
    pass
```

Exception handling

```
class ShortInputException(Exception):
    '''Exception raised if too short input.'''
    def __init__(self, length, atleast):
        '''Initialize.'''
        self.length = length
        self.atleast = atleast

try:
    s = raw_input('Enter something')
    if len(s) < 3:
        raise ShortInputException(len(s), 3)
except ShortInputException, ex:
    print 'Input was %d long, should be atleast %d' % (ex.length,
ex.atleast)
```

Command line input



Python frameworks / toolkits

- Standard libraries are extensive
([xml/os/network/compression/regex](#))
- GUI development - bindings to most GUI toolkits
([win32](#), [wxPython](#), [Tk](#), [Qt](#), [GTK](#), [Fox](#), ...)
- Web Application development frameworks
([Zope](#), [cherrypy](#), [quixote](#), [wsgi-paste](#), [mod_python](#))
- Network Programming - **Twisted**
- Scientific computing - **SciPy** ; Stats : **r-python**
- Machine Learning - **Orange**
- You want .NET and Java? Eh! **ironpython** and **jython**

Thank You!!



Slides available at:

<http://btbytes.com/slides/gat>

My blog:

<http://btbytes.com>

Email:

pradeep@btbytes.com

pradeep@mindscapecomputing.com

BangPypers

<http://www.bangpypers.org>

bangpypers@yahoogroups.com

- 2nd largest python group in the world.
- Monthly group meetings
- next meeting: 21st May
- I'm talking on wxPython GUI toolkit