Python for Beginners

Syntax and Libraries

introduction to Python

Python ?

- Object-oriented, high-level, interpreted language.
- Dynamic typing, dynamic binding.
- Modules and packages - modularity.
- Can be used from a script file or the Python shell.
- Popular IDEs: Visual Studio, PyCharm

Let’s get going !

```
import os
from numpy import random
from matplotlib import pyplot as
plt

name = ‘randomname’
anothername =  ‘namerandom’
variable,var =  17,97

# this line is a comment
print ”Hello World !”

“””
This is a multiline comment
”””
print variable * var
print name + anothername
```

- Keywords
- Identifiers
- Literals
- Punctuators
- Operators

Conditionals and Loops

```
for i in xrange(1,10,2):
        # starts from 1
        # ends at 9 (10 is NOT included)
        # moves in steps of 2
        print i,

a = 1
while a<11:
        print 2*a,

names = [‘priyansh’,’daksh]’
for name in names:
        if len(name) < 8: print name
        elif len(name)==8: print name, “loki”
        else: print “Why so serious ?”
```

lists

```
# this declares a list
this_is_a_list = []

# this also declares a list
this_is_another_list = list()

# add an element at the end
this_is_a_list.append('element')

# insert an element before index 2
this_is_a_list.insert(2, 56)

# add all elements from a list to another list
this_is_another_list.extend(this_is_a_list)

# find length
print len(this_is_a_list)
```

Slicing Notation

```
# this finds the first three elements of a list
some_list = this_is_a_list[:3]

# this finds the last two
another_list = this_is_a_list[-2:]

# this excludes the last two
one_more_list = this_is_a_list[:-2]

# this takes all the elements at odd indices
just_one_more_list = this_is_a_list[1::2]

# this takes the 2nd, 5th and 8th element
i_promise_this_is_the_last_list =  this_is_a_list[2:9:3]
```

WARNING: Copying a list using '=' operator marks a reference only, modifying the copied list will also change the original list.

Functions

```
# use def to define a function
def square(x):
        return x**2

# you can return multiple values
def i_give_two_values(random_input):
        return random_input[1], random_input[2] * 3

# using kwargs
# usage: print kwargs_are_lubb(player='Sachin', game='Cricket')
def kwargs_are_lubb(game, player):
        return game

# default arguments
def def_args(fruit, taste='sweet'):
        return fruit, taste
```


Pandas
Matplotlib
NumPy
SciKit
Data Science
DeZyre

NumPy

- **Fundamental package for scientific computing with Python.**
  - a powerful N-dimensional array object
  - sophisticated (broadcasting) functions
  - tools for integrating C/C++ and Fortran code
  - useful linear algebra, Fourier transform, and random number capabilities
- **NumPy’s main object - homogeneous multidimensional array.**

Let’s get going !

- Attributes of numpy array
  - ndim: the number of axes (dimensions) of the array.
  - shape: the dimensions of the array.
  - size: the number of elements in the array.
  - dtype: the datatype of the elements.
  - data: the buffer containing the elements of the array.

Examples

Examples on GitHub

MatplotLib

- **Fundamental package for scientific computing with Python.**
- **Can be used in Python scripts, the Jupyter notebook and web application servers.**
- **Can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots**

Examples

Examples on GitHub

Some useful modules/frameworks

- **AstroPy**
- **Jupyter Notebook**
- **IPython**

“The joy of coding Python should be in seeing short, concise, readable classes that express a lot of action in a small amount of clear code -- not in reams of trivial code that bores the reader to death.”

- **Guido van Rossum (author of Python, BDFL)**

Thanks!