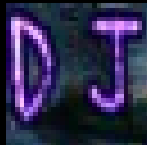




Anomit Ghosh  
anomit@anomit.com



Dhananjay Singh  
mail@djsh.net

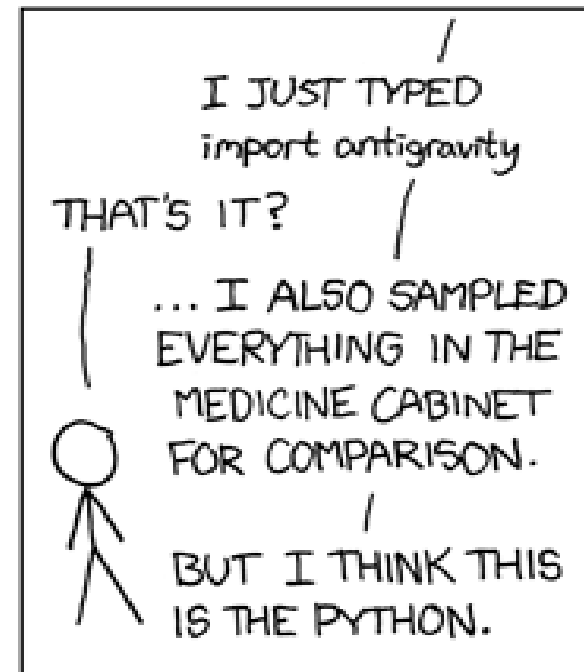


Ankur Shrivastava  
ankur@ankurs.com



# What We Will Learn

- History, Features and Basic Details
- Language Basics
- Control flow
- Functions
- Modules
- Data structures
- File I/O



# What is Python?

- Python is a general purpose, object oriented, high level, interpreted language
- Developed in early 90's by Guido Van Rossum
- Its simple, Portable, Open Source and Powerful

# Where is it used?

- Google uses it for its web crawler and search engine
- Youtube is based on Python
- Disney uses it in Panda3d
- Bittorrent is implemented in Python
- Used extensively from space shuttles to mobile phones

# Diff from C/C++/Java

- No Pointers (similar to java)
- No prior Compilation to ByteCode(?), directly Interpreted
- Includes Garbage Collector(?)
- Can be used in Procedure(?)/Object(?) Oriented approach/style
- Very good for scripting

# Versions Of Python

- What Do You mean By Versions??

Major/Important Versions Currently

- Python 2.5
- Python 2.6
- Python 3.0

So what will we use??

Why 2.5/2.6 and !3.0???

- Available on a Large Number of Systems
- Major Library's and Extensions available for 2.5/2.6 not 3.0
- <<<<MORE>>>>

# Python Interpreter

Interactive session

- What is it?
- Their Importance
- How to exit an Interactive session  
quit() or  
Ctrl + D on Unix-like systems  
Ctrl + Z on windows

# Language Basics

# Indentation

- Indentation is very important
- There are no begin/end delimiters
- Physical lines & Logical lines
- Joining physical lines by /
- Use # for comments

# Data Types

- Integer Numbers  $\rightarrow$  (dec) 1, 44;  
(oct) 01, 022; (hex) 0x1, 0x23;  
(long) 1L, 344343L
- Floating Point  $\rightarrow$   
0., 0.0, 323.23, 2e3, 23e32
- Complex numbers  $\rightarrow j = (-1)^{1/2}$   
10+10j, -1+5j, 5-6j

# Strings

- Strings can be single (' , " ) or triple ( ' ' ' , " " " ) quoted
- Character escaping by \
- Escape sequence - \\ , \' , \",  
\n , \t

# Tuple

- It is an Immutable(?) ordered sequence of items(?)
- Assigned  $\rightarrow a = (1122, 1212, 1212)$
- Using tuples  $\rightarrow$  can be used as a constant array (but are much more)
- Data can be accessed similar to an array  $\rightarrow a = (132, 3232, 323)$   
 $a[1]$  or  $a[3]$

# Lists

- List is a mutable ordered sequence of items (similar to tuple)
- Assigned  $\rightarrow a = [121, 121212, 34367]$
- Using Lists  $\rightarrow$  simplest use is a arrays (but again are much more)
- Data can be accessed similar to an array  $\rightarrow a = [132, 3232, 323]$   
 $a[1]$  or  $a[3]$

# Dictionaries

- Dictionaries are containers, which store items in a key/value pair(?)
- Assigned  $\rightarrow d = \{ 'x': 24, 'y': 33 \}$
- Using Dict  $\rightarrow$  They are used at a lot of places (covered later)
- Data can be accessed by using the key  $\rightarrow$   
 $d[ 'x' ]$

# Variables

- There is no prior declaration needed
- Variables are the references(?) to the allocated memory(?)
- Variables can refer to any data type (like Tuple, List, Dictionary, Int, String, Complex)
- References are share
- List, Dict etc are always shared

# Index and slices

- String, List, Tuple, etc can be sliced to get a part of them
- Index  $\rightarrow$  similar to array index, it refers to 1 position of data
- Slices  $\rightarrow$  gives the data in the range
- Example  $\rightarrow$

`a="LUG Manipal"`

`a[:3]`   `a[4:11]`   `a[4:]`   `a[-7:]`   `a[:-8]`   `a[:11:2]`

Control Flow

# print

- Print is a simple statement for giving output similar to C's printf function
- Can be used to output to Console or a file
- Use `→ print "Hello World"`

# input

- Use `raw_input()` to take a string input from the user
- Used as  
`<var> = raw_input("Enter a string: ")`
- `Input()` is used to take an input without specifying the type

# if

- If is a conditional statement, for simple "If then else" clause in English
- Header lines(?) are always concluded with a " : " followed by intended block of statements
- Optionally it can be followed by an "else if" clause known as "elif" in python

# If (con)

- Example →
- if <condition>:  
    statement 1  
    statement 2
- elif <condition>:  
    statements
- else:  
    statements

# while

- While statement is used for repeatedly executing a block of code till the condition is true, also has an optional else clause
- Use wildy for infinite loop

# While (con)

- Example →
- While <condition>:  
    statements  
else:  
    statements

# for

- It is a sequence iterator(?)
- It works on strings, lists, tuples, etc
- Example →
- For <target> in <iterable>:  
    statements

# range/xrange

- They are used to generate and return integer sequence(?)
- Range(5)  $\rightarrow$  [0,1,2,3,4]
- Range(1,5)  $\rightarrow$  [1,2,3,4]
- Range(0,8,2)  $\rightarrow$  [0,2,4,6]
- Xrange is used as a memory efficient(?) alternative for range

# break

- Used to terminate a loop
- If nested(?) it terminates the inner most loop
- Practically used for conditional loop termination with an if statement

# continue

- Terminates the current iteration and executes next
- Practically used for conditional statements termination with an if statement

# Some Helpful Functions

- `Dir()`
- `Help()`

Functions

# What are functions?

- A Function is a group of statements that execute on request
- In Python Functions are Objects
- Defining a function →  

```
def name(parameters):  
    statement(s)
```

# Parameters

- Types of parameters

  - Mandatory Parameters

  - optional parameters

- Defaults values

- Be careful when default value is a mutable object

# Example

- Def `a(x, y=[]):`

```
    y.append(x)
```

```
    print y
```

```
print a(12)
```

```
print a(34)
```

- What just happened here?

# Namespace

- A namespace is an abstract container or environment created to hold a logical grouping of unique(!) identifiers or symbols
- <<MORE>>

# Nested Functions

- Def statement inside a function defines a nested function
- Useful sometimes
- Use of Class preferred

```
def getSolutionCosts (navigationCode):
```

```
    fuelStopCost = 15
```

```
    extraComputationCost = 8
```

```
    thisAlgorithmBecomingSkynetCost = 999999999
```

```
    waterCrossingCost = 45
```



GENETIC ALGORITHMS TIP:

*ALWAYS* INCLUDE THIS IN YOUR FITNESS FUNCTION

# Functions (con)

- Return
- Return Type
- Some Examples

Modules

# Modules

- What are modules?
- How to load modules
- Effect on namespace
- Important modules

OS

- SYS

How To Make Modules?

Y is this Important

