Python

Submitted by:

Shivam Gupta(1318710097)

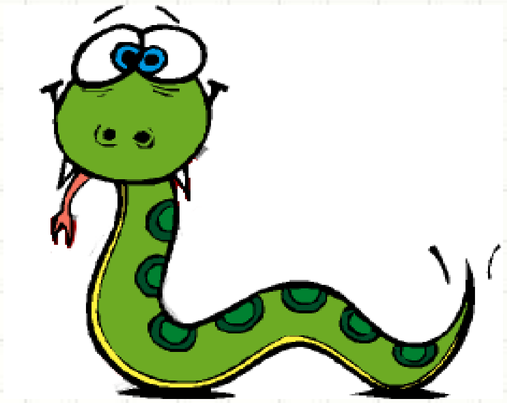
Shashendra Singh(1318710094)

What We Give you?

- What is Python…?
- Differences between program and scripting language
- History of Python
- Scope of Python
- What can I do with python
- Who uses python today
- Why do people use Python?
- Installing Python IDE
- A Sample Code
- Python code execution
- Running Python
- Python Basic(Variable, Strings, Data types etc.)

What is Python…?

- Python is a general purpose programming language that is often applied in scripting roles.
- So, Python is programming language as well as scripting language.
- Python is also called as Interpreted language



Differences between program and scripting language

Program

- a **program is executed** *(i.e. the source is first compiled, and the result of that compilation is expected)*
- A "program" in general, is **a sequence of instructions written so that a computer can perform certain task**.

Scripting

- a **script is interpreted**
- A "script" is code written in a scripting language. A scripting language is nothing but **a type of programming language in which we can write code to control another** software application.

History

- Invented in the Netherlands, early 90s by Guido van Rossum
- Python was conceived in the late 1980s and its implementation was started in December 1989
- Guido Van Rossum is fan of ‘**Monty Python’s Flying Circus**’, this is a famous TV show in Netherlands
- Named after Monty Python
- Open sourced from the beginning

Python's Benevolent Dictator For Life

"Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered."

- Guido van Rossum

Why was python created?

"My original motivation for creating Python was the perceived need for a higher level language in the Amoeba [Operating Systems] project.

I realized that the development of system administration utilities in C was taking too long. Moreover, doing these things in the Bourne shell wouldn't work for a variety of reasons. ...

So, there was a need for a language that would bridge the gap between C and the shell”

- Guido Van Rossum

Scope of Python

- Science

  - Bioinformatics
- System Administration

  -Unix

  -Web logic

  -Web sphere
- Web Application Development

  -CGI

  -Jython – Servlets
- Testing scripts

What can I do with Python…?

- System programming
- Graphical User Interface Programming
- Internet Scripting
- Component Integration
- Database Programming
- Gaming, Images, XML , Robot and more

Who uses python today…

- Python is being applied in real revenue-generating products by real companies. For instance:
- Google makes extensive use of Python in its web search system, and employs Python’s creator.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.
- ESRI uses Python as an end-user customization tool for its popular GIS mapping products.
- The YouTube video sharing service is largely written in Python

Why do people use Python…?

The following primary factors cited by Python users seem to be these:

- Python is object-oriented

  Structure supports such concepts as polymorphism, operation overloading, and multiple inheritance.

.

- It's free (open source)

  Downloading and installing Python is free and easy

  Source code is easily accessible

- It's powerful
  - Dynamic typing
  - Built-in types and tools
  - Library utilities
  - Third party utilities (e.g. Numeric, NumPy, SciPy)
  - Automatic memory management
- It's portable
  - Python runs virtually every major platform used today
  - As long as you have a compatible Python interpreter installed, Python programs will run in exactly the same manner, irrespective of platform.

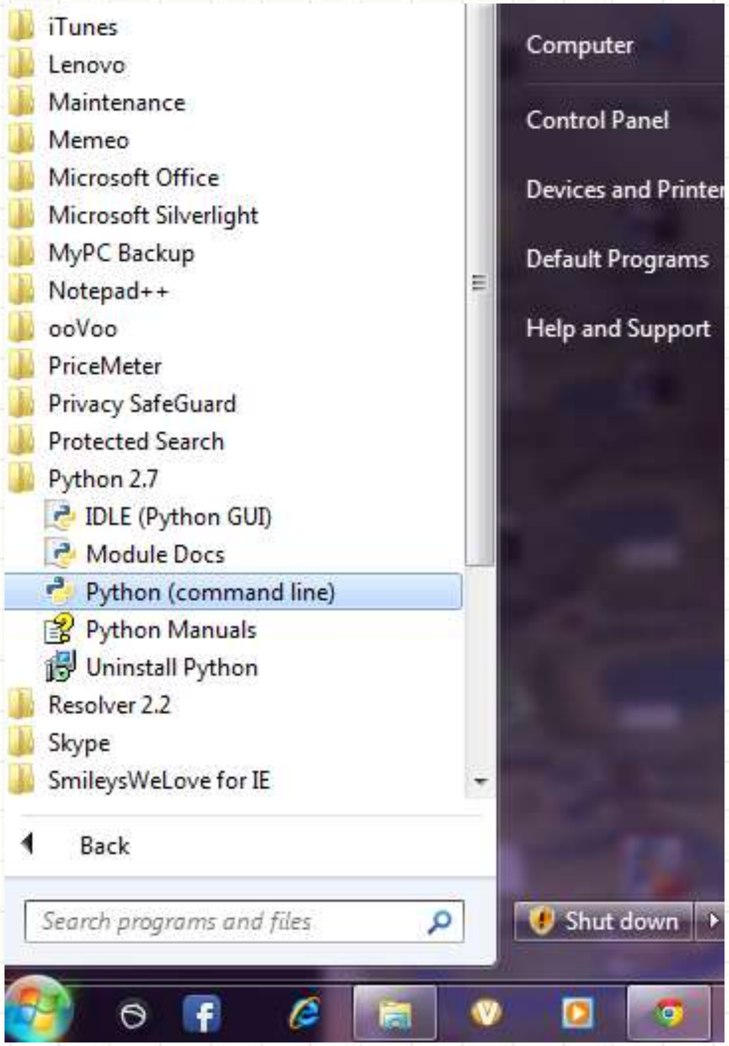
Installing Python

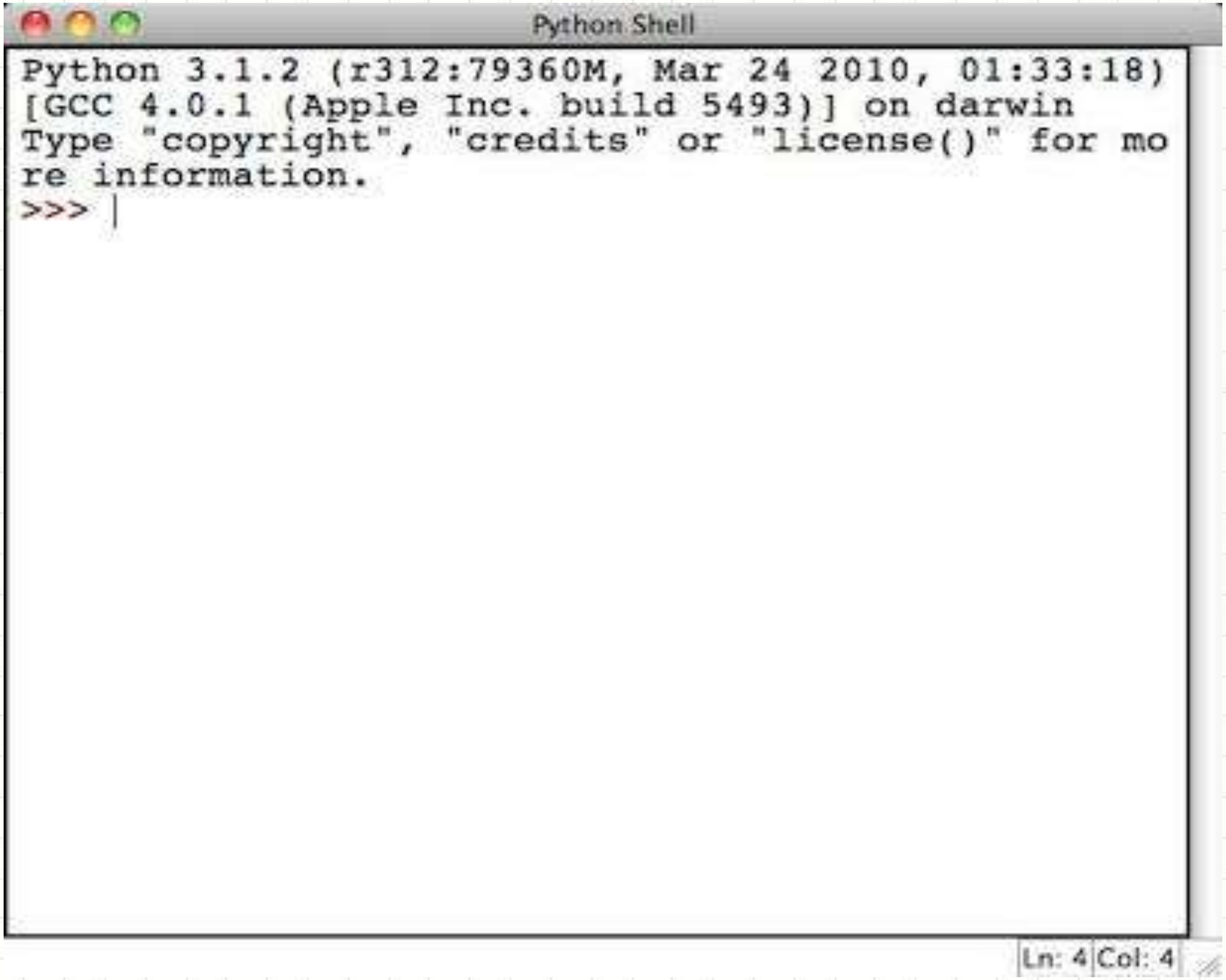
- Python is pre-installed on most Unix systems, including Linux and MAC OS X

- But for in Windows Operating Systems , user can download from the https://www.python.org/downloads/

  - from the above link download latest version of python IDE and install, recent version is 3.4.1 but most of them uses version 2.7.7 only

- After installing the Python Ver#2.7.7, go to start menu then click on python 2.7 in that one you can select python (command line) it is prompt with >>>





```
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for mo
re information.
>>> 
```

Running Python

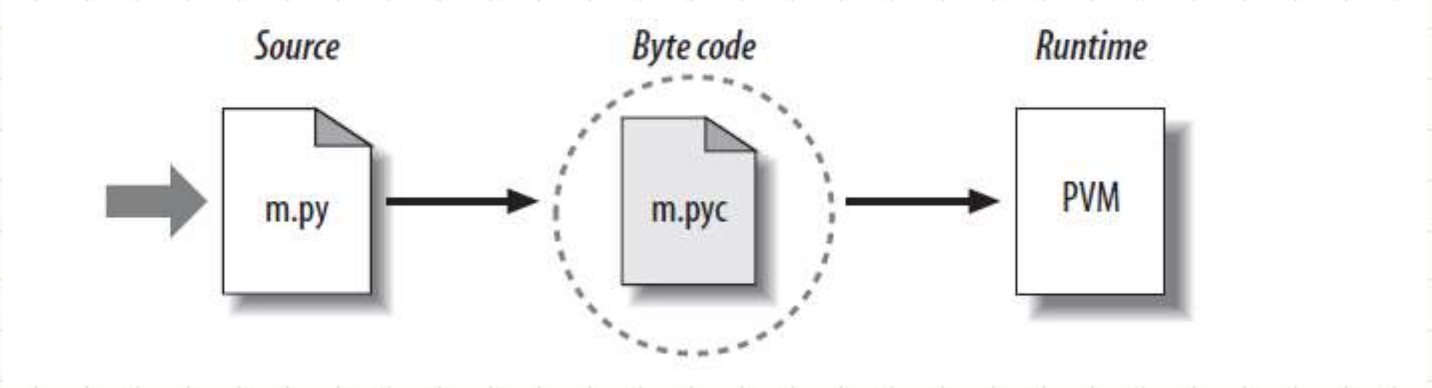
Once you're inside the Python interpreter, type in commands at will.

- Examples:

```
>>> print 'Hello world'
    Hello world
```

Python Code Execution

- Python's traditional runtime execution model: source code you type is translated to byte code, which is then run by the Python Virtual Machine. Your code is automatically compiled, but then it is interpreted.



Source code extension is **.py**
Byte code extension is **.pyc** (compiled python code)

Math(Operator) In Python

Math

Try typing this into Code:

```
>>> print 3 + 12
      15
>>> print 12 – 3
      9
>>> print 9 + 5 – 15 + 12
     11
```

<u>Operators:</u>

add: +

subtract: -

<u>Note</u>: don’t type the arrows >>> !

Math

Rule: If you want Python to answer in floats, you have to talk to it in floats.

More operators:

divide: /

multiply: *

```
>>> print 3 * 12         36
>>> print 12 / 3         4
>>> print 11 / 3         3
>>> print 12.0 / 3.0     4.0
>>> print 11.0 / 3.0     3.66
```

Math

Practice:

```
>>> print 2 < 3                   True
>>> print 2 <= 2                  False
>>> print 3 > 2                   True
>>> print 2 != 3                  True
>>> print False < True            True
```

Strings In Python

Strings

Examples:

```
>>> "It's a beautiful day!"
>>> "Goodbye, cruel world."
```

Try typing one without quotes:

What's the result?

```
>>> Aggies

>>> "Aggies"
>>> "Rice fight, never die!"
>>> "3 + 2"
```

Strings

String operators:

concatenation: +

multiplication: *

Try concatenating:

Try multiplying:

```
>>> print "Hello" + " " + "world!"
>>> print "HAHA" * 250
```

Variables In Python

Variable

Create a Variable:

```
>>>headmaster=“Dumbledore”
>>>print headmaster
    ‘Dumbledore’
```

Assigning a New Value:

```
>>>headmaster=“Hardcastle”
>>>print headmaster
   ‘Hardcastle’
```

Data types In Python

Data Type:

Python has many native data types. Here are the important ones:

Booleans are either True or False.

Numbers can be integers (1 and 2), floats (1.1 and 1.2), fractions (1/2 and 2/3), or even complex numbers.

Strings are sequences of Unicode characters, e.g. an HTML document.

Bytes and byte arrays, e.g. a JPEG image file.

Lists are ordered sequences of values.

Tuples are ordered, immutable sequences of values.

Sets are unordered bags of values.

Example:

String	`“Whoop!”`
Integer	42
Float	`3.14159`
List	[“John”, “Paul”, “George”, “Ringo”]

Python can tell us about types using the type() function:

```
>>> print type(“Whoop!”)
<type ‘str’>
```

List: Data Type

List:

The list is a most versatile Data type available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Example:

```
list1 = ['physics', 'chemistry', 1997, 2000];
list2 = [1, 2, 3, 4, 5 ];
```

SN	Function with Description
1	**cmp(list1, list2)** Compares elements of both lists.
2	**len(list)** Gives the total length of the list.
3	**max(list)** Returns item from the list with max value.
4	**min(list)** Returns item from the list with min value.
5	**list(seq)** Converts a tuple into list.

List: a sequence of objects

```
>>> Beatles = [“John”, “Paul”, “George”,
“Ringo”]
>>> grades = [82, 93, 67, 99, 100]
```

Guess what this will output:

```
>>> type(Beatles)


>>> type(grades)
```

Lists

Index: Where an item is in the list

```
>>> Beatles = ["John", "Paul", "George", "Ringo"]
>>> Beatles[0]
'John'
```

```
["John", "Paul", "George", "Ringo"]
     0        1          2           3
```

Python always starts at zero!

Tuple: Data Type

Tuples:

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

Example:

```
tup2 = (1, 2, 3, 4, 5 );
tup3 = ("a", "b", "c", "d“);
```

Accessing Values:

```
print "tup2[1:5]: “
```

Output:

```
tup2[1:5]:  [2, 3, 4, 5]
```

Built-in Tuple Functions

Python includes the following tuple functions –

SN	Function with Description
1	**cmp(tuple1, tuple2)** Compares elements of both tuples.
2	**len(tuple)** Gives the total length of the tuple.
3	**max(tuple)** Returns item from the tuple with max value.
4	**min(tuple)** Returns item from the tuple with min value.
5	**tuple(seq)** Converts a list into tuple.

Loops & Conditional Statements

Loop Type	Description
while loop	Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
for loop	Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
nested loops	You can use one or more loop inside any another while, for or do..while loop.

Statement	Description
if statements	An **if statement** consists of a boolean expression followed by one or more statements.
if...else statements	An **if statement** can be followed by an optional **else statement**, which executes when the boolean expression is FALSE.
nested if statements	You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).

I believe the trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language:

- It is Free (as in both cost and source code).
- It is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated;
- It is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills;
- It is a real-world programming language that can be and is used in academia and the commercial world;
- It appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied;
- and most importantly, its clean syntax offers increased understanding and enjoyment for students;

Thank You!