# Learn
# Python Programming

# What is an Algorithm and Flow Chart

- **Algorithm** is a sequence of activities to be processed for getting desired output from given input
- A step by step method for solving a particular problem or doing a task
- **Flowchart** is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows
- We can see a flow chart as a blueprint of a design you have made for solving a problem.

# What is Programming Language

- Programming Language is also like an English,Telugu…etc.
- A vocabulary and set of grammatical rules for instructing a computer to perform specific tasks.
- Each language has a unique set of keywords and a special syntax for organizing program instructions.
- Programming languages are classified as:

  Low level language and High level language
- The term **programming language** usually refers to high-level **languages**,

  such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal.

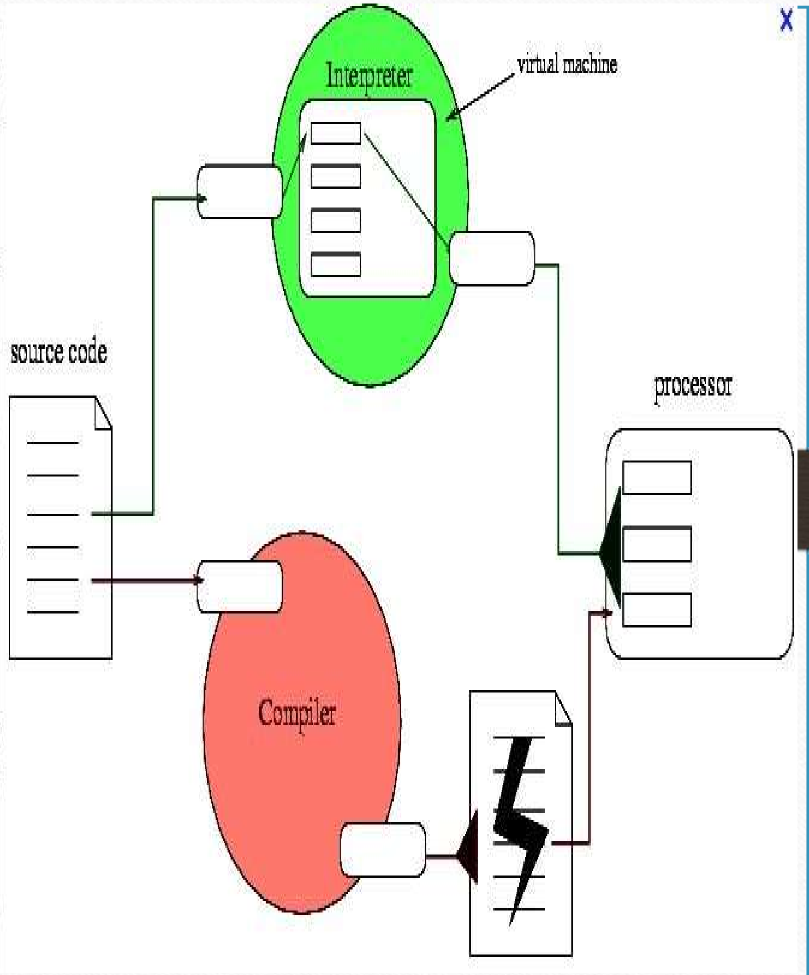
- **LOW LEVEL LANGUAGE:**
  - **MACHINE LANGUAGE :-** the language of 0s and 1s is called as machine language. The machine language is system independent because there are different set of binary instruction for different types of computer systems
  - **ASSEMBLY LANGUAGES:-** it is low level programming language in which the sequence of 0s and 1s are replaced by mnemonic (ni-monic) codes. Typical instruction for addition and subtraction .
    - **Example :-** ADD for addition , SUB for subtraction etc
- Since our system only understand the language of 0s and 1s . therefore a system program is known as assembler . Which is designed to translate an assembly language program into the machine language program.
- **HIGH LEVEL LANGUAGE :-** high level languages are English like statements and programs . Written in these languages are needed to be translated into machine language before to their execution using a system software compiler
- Program written in high level languages are much easier to maintain and modified .
- High level language program is also called source code.
- Machine language program is also called object code.

# What is Interpreter and Compiler

- We generally write a computer program using a high-level language. A high-level language is one which is understandable by us humans. It contains words and phrases from the English (or other) language.
- But a computer does not understand high-level language. It only understands program written in 0's and 1's in binary, called the machine code.
- A program written in high-level language is called a source code. We need to convert the source code into machine code and this is accomplished(actioned) by compilers and interpreters.
- Hence, a compiler or an interpreter is a program that converts program written in high-level language into machine code understood by the computer.

# Note

- Interpreted languages do not need to be **compiled** to run. A program called an **interpreter** will run python code on any kind of computer it can run on itself.
- This means if the programmer needs to change the code they can quickly see the results.
- This also means Python is slower than a compiled language like C, because it is not running **machine code** directly.



| Interpreter | Compiler |
|---|---|
| **Translates program one statement at a time.** | Scans the entire program and translates it as a whole into machine code. |
| **It takes less amount of time to analyze the source code but the overall execution time is slower.** | It takes large amount of time to analyze the source code but the overall execution time is comparatively faster. |
| **Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.** | It generates the error message only after scanning the whole program. Hence debugging is comparatively hard. |
| **Programming language like Python, Ruby use interpreters.** | Programming language like C, C++ use compilers. |

# What is Python

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics(substance).
- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse
- Python is an **open source programming language** made to both look good and be easy to read.

# History of Python

- Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 20, 1991.
- **Why Python was created?**
  - In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group.
  - He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls.
  - So, he decided to create a language that was extensible. This led to a design of new language which was later named Python.
  - Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

## Why the name was Python?

- No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series Monty Python's Flying Circus in late seventies.
- The name "Python" was adopted from the same series "Monty Python's Flying Circus".

## Release Dates of Different Versions

| Version | Release Data |
|---|---|
| Python 1.0 (first standard release)<br>Python 1.6 (Last minor version) | January 1994<br>September 5, 2000 |
| Python 2.0 (Introduced list comprehensions)<br>Python 2.7 (Last minor version) | October 16, 2000<br>July 3, 2010 |
| Python 3.0 (Emphasis on removing duplicative constructs and module)<br>Python 3.7 (Last updated version) | December 3, 2008<br>June 27, 2018 |

# Features of Python Programming

### A simple language which is easier to learn

- Python has a very simple and elegant (graceful) syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#.

### Free and open-source

- You can even make changes to the Python's source code and update.

### Portability

- You can move Python programs from one platform to another, and run it without any changes. It runs smoothly on almost all platforms including Windows, Mac OS X and Linux.

### Extensible and Embeddable

- Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code and other languages may not provide out of the box

## A high-level, interpreted language

- Unlike C/C++, you don't have to worry about daunting(Cause to lose courage) tasks like memory management, garbage collection and so on, likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

## Large standard libraries to solve common tasks

- Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself
- **For example:** Need to connect MySQL database on a Web server? You can use MySQLdb library using ***import MySQLdb***

## Object-oriented

- Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem **intuitively**.
- With OOP, you are able to divide these complex problems into smaller sets by creating objects.

# Why Python is very easy to read Language ?

- One big change with Python is the use of whitespace to define code: spaces or tabs are used to organize code by the amount of spaces or tabs.
- This means at the end of each line, a **semicolon** is not needed and curly braces ({}) are not used to group code.
- Which are both common in C. The combined effect makes Python a very easy to read language.

# 4 Reason to Choose Python as First Language

## Simple Elegant (Graceful) Syntax

- It's easier to understand and write Python code.
- Why ?Syntax feels Naturals With Example Code

```
A=12
B=23
sum=A+B
print(sum)
```

- Even if you have never programmed before, you can easily guess that this program adds two numbers and prints it.

## Not overly strict

- You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.
- Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

## Expressiveness of the language

- Python allows you to write programs having greater functionality with fewer(less) lines of code.
- we can build game**(Tic-tac-toe)** with Graphical interface in less than 500 lines of code
- This is just an example. You will be amazed how much you can do with Python once you learn the basics

## Great Community and Support

- Python has a large supporting community.
- There are numerous active forums online which can be handy if you are stuck

# Run Python on Your Operating System

- **Install and Run Python in Linux (Ubuntu)**

1. Install the following dependencies:
    - $ sudo apt-get install **build-essential** checkinstall
    - $sudo apt-get install **libsqlite3-dev**
    - $ sudo apt-get install **libbz2-dev**

    **(libreadline-gplv2-dev** libncursesw5-dev **libssl-dev** tk-dev **libgdbm-dev** libc6-dev)

    Video Link : - https://www.youtube.com/watch?v=sKiDjO_0dCQ
2. Go to Download Python page on the official site and click **Download Python 3.7.0**
3. In the terminal, go to the directory where the file is downloaded and run the command:
    - $ tar -xvf Python-3.7.0.tgz

- This will extract your zipped file.
- Note: The filename will be different if you've downloaded a different version. Use the appropriate filename

4. Go to the extracted directory.
   - $ cd Python-3.7.0
5. Issue the following commands to compile Python source code on your Operating system
   - $ ./configure
   - $ sudo make altinstall
6. Go to Terminal and type the following to run sample ‘ hello world ‘ Program

```
ubuntu@RGUKT~$ python3.7.0
>>>print(‘Hellow World’)
Hello World
>>>
```

## 4. To set default version as new python series

- Open your .bashrc file by typing in terminal
- ubuntu@RGUKT~$ **nano ~/.bashrc**.

Then type **alias python=python3** on to a new line at the top of the file then **save the file with ctrl+o** and **close the file with ctrl+x**.

Then, back at your command line **type**

```
 ubuntu@RGUKT~$ source ~/.bashrc.
```

Now your alias should be permanent.

Go to Terminal and type python instead of python3.7 the following to run

```
ubuntu@RGUKT~$ python
```

# Install and Run Python in Windows

1. Go to Download Python page on the official site and click Download Python 3.4.3
2. When the download is completed, double-click the file and follow the instructions to install it
3. When Python is installed, a program called IDLE is also installed along with it. It provides graphical user interface to work with Python
4. Open IDLE, copy the following code below and press enter
    - print("Hello, World ")
5. To create a file in IDLE, go **to File > New Window** (Shortcut: **Ctrl+N**).
6. Write Python code (you can copy the code below for now) and save (Shortcut: **Ctrl+S**) with **.py** file extension like: **hello.py** or **your-first-program.py**
    - print("Hello, World ")
7. Go to Run > Run module (Shortcut: **F5**) and you can see the output. Congratulations, you've successfully run your first Python program

# Type of Modes Python in Windows

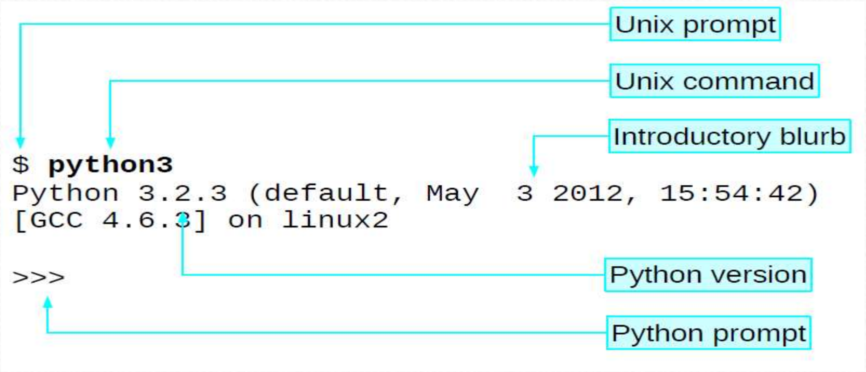
- There are various ways to start Python
- Immediate Mode or Interactive Mode
  - Typing python in the command line will invoke the interpreter in immediate mode. We can directly type in Python expressions and press enter to get the output **(>>>)**
  - Is the python prompt and it tells us interpreter is ready for input
- Script Mode
  - This mode is used to execute Python program written in a file. Such a file is called a script. Scripts can be saved to disk for future use. Python scripts have the extension .py, meaning that the filename ends with **.py**.
  - For example: **helloWorld.py**
  - To execute this file in script mode we simply write python helloWorld.py at the command prompt.
- **Integrated Development Environment (IDE)**
  - We can use any text editing software to write a Python script file. Like Notepad, Editplus,sublime…etc

# Running Python

- Two different ways
  - Interactive Interpreter
  - Script from the Command-line

# Interactive Interpreter

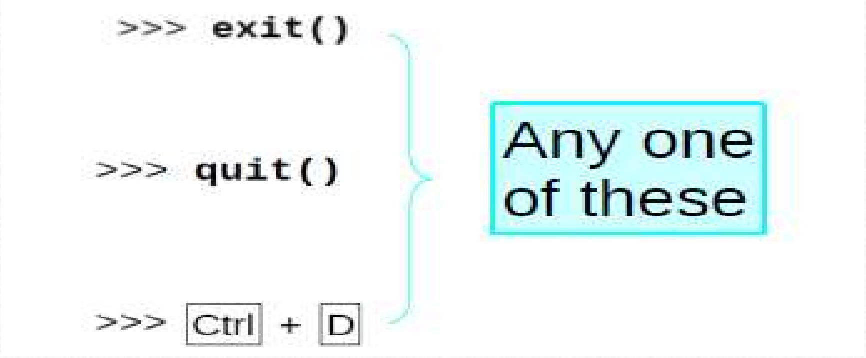
- Enter **python3.7** the command line



```
$ python3
Python 3.2.3 (default, May  3 2012, 15:54:42)
[GCC 4.6.3] on linux2

>>>
```

- Quitting Python



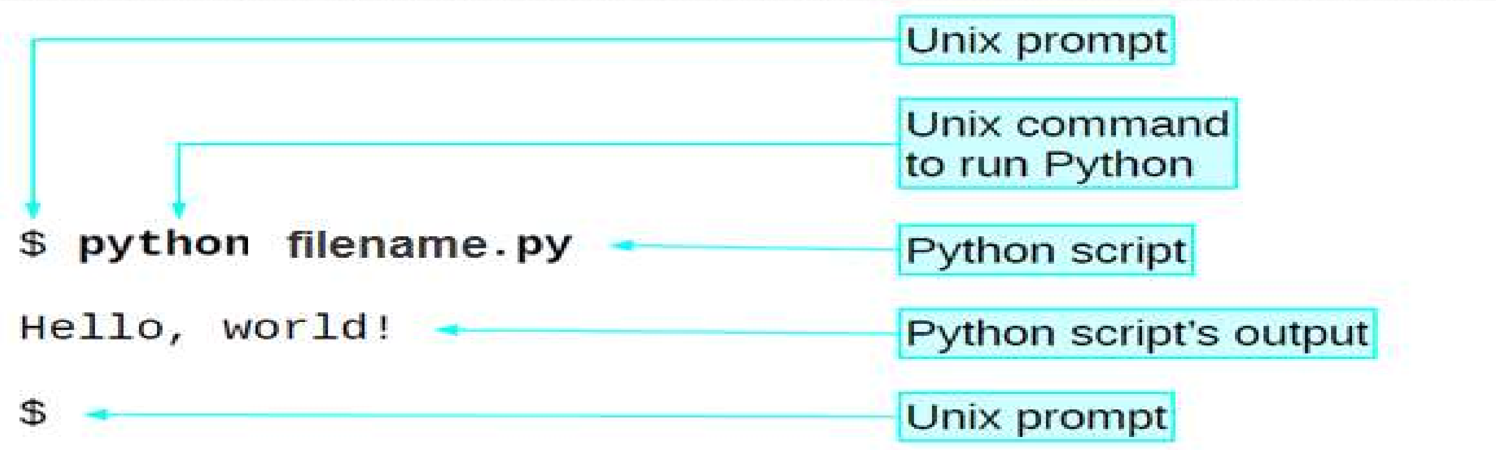
```
>>> exit()

>>> quit()

>>> Ctrl + D
```

# Script from the Command-line

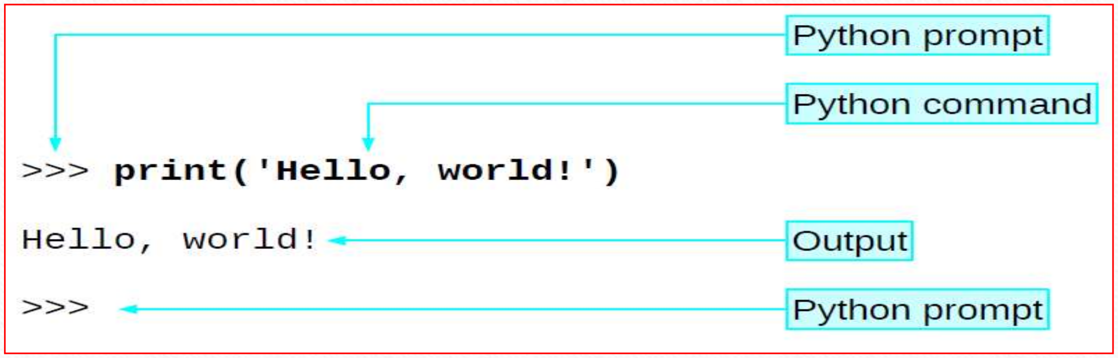
- `$python3.7 fileName.py`



```
$ python filename.py
Hello, world!
$
```

# First Program in Python: Printing a Line of Text

- Interactive mode



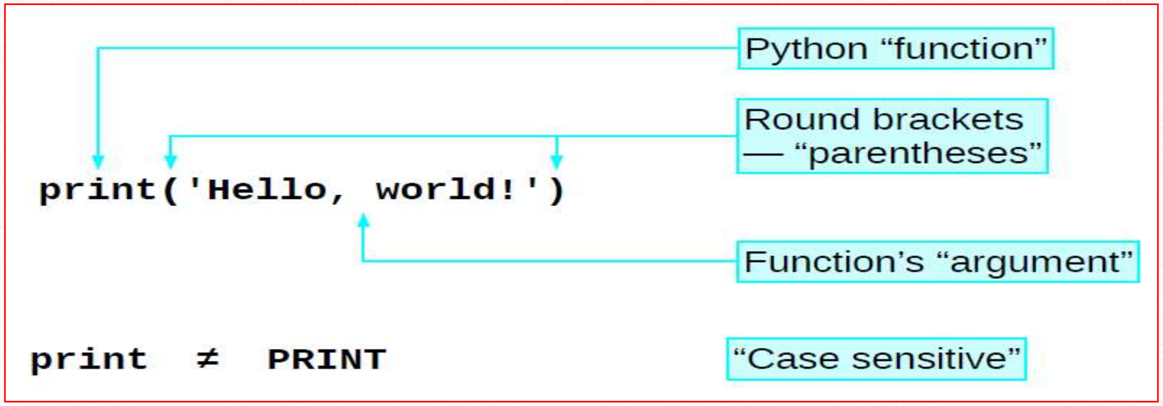
```
>>> print('Hello, world!')
Hello, world!
>>>
```

Python prompt

Python command

Output

Python prompt



```
print('Hello, world!')

print  ≠  PRINT
```

Python "function"

Round brackets — "parentheses"

Function's "argument"

"Case sensitive"

# First Program in Python: Printing a Line of Text

- Script Mode
  - Saving as a file
    - Type code into a .py file and save it
    - To run it type **`python`** ***`fileName`*** **`.py`**

```
# hello.py
# Printing a line of text in Python
print("Welcome to Python!")
```

Comments

Printing Line

Output of the above program will be

**Welcome to Python!**