Day 2 :
Topic coverd : Introduction of Python

High-level language vs low-level language

High-level programming languages are those which are programmer-friendly. Humans can easily understand it which helps in easy maintaining and debugging of the code. These types of languages need a compiler or interpreter for execution. High-level languages are the most commonly used languages. Examples – Java, C, C++, Python

Low-level programming languages are those which are machine-friendly. Humans can't understand such languages. In other words, low-level languages are not human-readable. So these languages, comparatively, are complex to debug. An assembler is required to debug these types of languages. Examples – Assembly language, Machine language.

Compiler vs Interpreter

All the programming languages are converted to machine understandable language for execution. The conversion is taken care of by compilers and interpreters.

A compiler is one that takes the source code and converts it into machine-executable code at once. The processor takes the executable and executes it. The languages which use a compiler are called compiled languages. Compiled languages are faster and more efficient comparatively. Examples – Java, C++

An interpreter is one that runs the code line by line and executes instruction by instruction. The languages which use interpreters are called Interpreted languages. Such languages are slower in terms of execution and less efficient. Examples – PHP, Python, Ruby

Scripting language

A scripting language is a programming language that is interpreted. These types of languages are used in automation, etc. These languages use interpreters and are executed line by line.

Definition of Python:

Python is a general-purpose, interpreted, high-level programming language, a Wikipedia definition. In clear terms, Python

High-level language – It is a human-readable language which easily understandable and easy to debug

Interpreted as well as compiled – Python is basically called an interpreted language, but can also be called a compiled interpreted language. In Python, first, all the source code is at once converted

to byte code which is the action of the compiler, and then the byte code is executed by the compiler.

Python is also called a scripting language.

History or Origin of Python

It was invented in the Netherlands in the early ’90s.

Guido Van Rossum was the creator of this beautiful language.

Guido released the first version in 1991.

Python was derived from the ABC programming language, a general-purpose programming language.

It is open-source software that can be downloaded freely and the code is customizable as well.

Why the name Python?

There was a TV show by the name Monty Python’s Flying Circus which was a very much popular fun show in the 1970s. While creating Python, Guido also used to read this show’s published scripts. Guido needed a short and a different name for this language, hence he named it “Python”.

Python Versions

Python 1.0V was introduced in Jan 1994

Python 2.0V was introduced in October 2000

Python 3.0V was introduced in December 2008.

The current version is 3.8

Note: Python 3 won’t provide backward compatibility to Python2 i.e. there is no guarantee that Python2 programs will run in Python3.

Features of Python

Simple: Python syntax is very easy. Developing and understanding python is very easy than others. The below comparison illustrated how simple python language is when compared to other languages.

Introduction to Python

Open Source – We can download freely and customize the code as well

Dynamically typed – Dynamically type will be assigned to data.

Platform independent: Python programs are not dependent on any specific operating systems. We can run on all operating systems happily.

Portable: If a program gives the same result on any platform then it is a portable program. Python used to give the same result on any platform.

Huge library – Python has a big library to fulfill the requirements.

Database connectivity – Python provides interfaces to connect with all major databases like oracle, MySQL

Batteries included – Python provides inbuilt libraries called batteries. Some of them are below

Boto – amazon web services library

MySQL-connector-python – To connect with MySQL

NumPy – To process arrays

Pandas – powerful data structures for data analysis, time series, and statistics

Where all python is used?

Python is being used to develop:

Standalone applications: An application that needs to install on every machine to work with that application.

Web applications: An application that follows a client-server architecture. The client is a program, which sends requests to the server. The server is a program, mainly it can do three things. Captures the request from the client, process the request, and Sends the response to the client

Database applications.

To process huge amounts of data. Ex Hadoop, Spark.

Machine learning.

Artificial Intelligence.

Data science.

Network servers.

IoT

Application scripting etc.

Python supports

Functional programming as well as Object-oriented programming approach.

Initial languages like C, Pascal, or FORTRAN follow functional approaches.

C++, Java, and dot net follow an object-oriented approach.

Python follows both functional and object-oriented approaches.

Python is an interpreted language

An interpreter is very different from a compiler. An interpreter executes the statements of code "one-by-one" whereas the compiler executes the code entirely and lists all possible errors at a time. That's why python shows only one error message even though your code has multiple errors. This will help you to clear errors easily.

Keywords in Python

All keywords in python contain only alphabet symbols. All of them are in lower case except True, False, and None. To see all the keywords –

import keyword;

keyword.kwlist

Keywords – ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Different flavors of python

Flavors of python refer to the different types of python compilers. These are useful to integrate various programming languages.

CPython: Standard python compiler implemented in C language. This is the python software being downloaded and used by the programmers directly.

Jython: Initially called as JPython, later renamed to Jython. Designed to run on a Java program.

IronPython: Designed for .NET framework.

PyPy: The main advantage of PyPy is performance will be improved because the JIT compiler is available inside PVM.

RubyPython: This is a bridge between Ruby and Python interpreters. It encloses a python interpreter inside the Ruby application.

AnacondaPython: Anaconda is a free and open-source Python programming language. This is mainly for data science and machine learning-related applications (large-scale data processing,

predictive analytics, scientific computing). This aims to simplify package management and deployment.

Applications of Python

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

Here, we are specifying application areas where Python can be applied.

Python Applications

1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser, etc. One of Python web-framework named Django is used on Instagram. Python provides many useful frameworks, and these are given below:

Django and Pyramid framework(Use for heavy applications)

Flask and Bottle (Micro-framework)

Plone and Django CMS (Advance Content management)

2) Desktop GUI Applications

The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a Tk GUI library to develop a user interface. Some popular GUI libraries are given below.

Tkinter or Tk

wxWidgetM

Kivy (used for writing multitouch applications)

PyQt or Pyside

3) Console-based Application

Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. This kind of application was more popular in the old generation of computers.

Python provides many free library or module which helps to build the command-line apps. The necessary IO libraries are used to read and write. It helps to parse argument and create console

help text out-of-the-box. There are also advance libraries that can develop independent console apps.

4) Software Development

Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.

SCons is used to build control.

Buildbot and Apache Gumps are used for automated continuous compilation and testing.

Round or Trac for bug tracking and project management.

5) Scientific and Numeric

This is the era of Artificial intelligence where the machine can perform the task the same as the human. Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations.

Implementing machine learning algorithms require complex mathematical calculation. Python has many libraries for scientific and numeric such as Numpy, Pandas, Scipy, Scikit-learn, etc. If you have some basic knowledge of Python, you need to import libraries on the top of the code. Few popular frameworks of machine libraries are given below.

SciPy

Scikit-learn

NumPy

Pandas

Matplotlib

6) Business Applications

Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features.

Oddo is an example of the all-in-one Python-based application which offers a range of business applications. Python provides a Tryton platform which is used to develop the business application.

7) Audio or Video-based Applications

Python is flexible to perform multiple tasks and can be used to create multimedia applications. Some multimedia applications which are made by using Python are TimPlayer, cplay, etc. The few multimedia libraries are given below.

Gstreamer

Pyglet

QT Phonon

8) 3D CAD Applications

The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.

Fandango (Popular)

CAMVOX

HeeksCNC

AnyCAD

RCAM

9) Enterprise Applications

Python can be used to create applications that can be used within an Enterprise or an Organization. Some real-time applications are OpenERP, Tryton, Picalo, etc.

10) Image Processing Application

Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.

OpenCV

Pillow

SimpleITK

Python Coding Instructions

Installation procedure For Windows:

You can download a python installer from https://www.python.org/downloads/ depending upon the operating system and the version (32-bit or 64-bit version) your system has. Download the latest stable version. (The current version is 3.x.x.x)

After downloading the installer from the above site you have to run the installer and follow the instructions in order to get the Python installed in your system. During the setup, window make sure to click on "Add Python 3.6 to PATH" or 'Add Python to your environment variables" checkbox before clicking on Install Now.

Note: Don't forget to check the box as mentioned above. If not checked then you have to add the Python to your environment variables.

Ways to write a python program

All the programs written in python should be saved using the '.py' or '.python' extension. Files can be executed using the command prompt or terminal (based on OS). For writing the python programs we can use the following things

By using a text editor like notepad++, VI Editor, etc…

You can use IDEs available online like VS Code, Atom, PyCharm, etc.

Python has a special feature of interactive mode where the python code can be executed line by line. In the command prompt/terminal if you type 'python' then the interactive mode comes as shown below

```
Python 3.12.0b3 (tags/v3.12.0b3:f992a60, Jun 20 2023, 12:25:40) [MSC v.1936 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>>exit()
```

Note: In interactive mode, we type the instructions line by line. exit() is used to quit from the interactive mode.

Python IDLE:

An IDLE is automatically installed while installing the python. If you have python installed on the system then search for it using 'IDLE'. You can use that IDLE for both interactive modes as well as

the normal mode of executing the python file using the command. For this tutorial, it is suggested to follow the below steps for execution.

Program execution steps

First write the code in notepad or IDE (Initially, it would be better to write the code in notepad if you are new to IDEs)

Save the file with .py or .python extension.

Run or execute the program (For windows execute the file from command prompt and for linux from terminal)

Command for execution – "python filename.py" or "py filename.py"

Program to print hello world

Code:

print("HELLO WORLD")

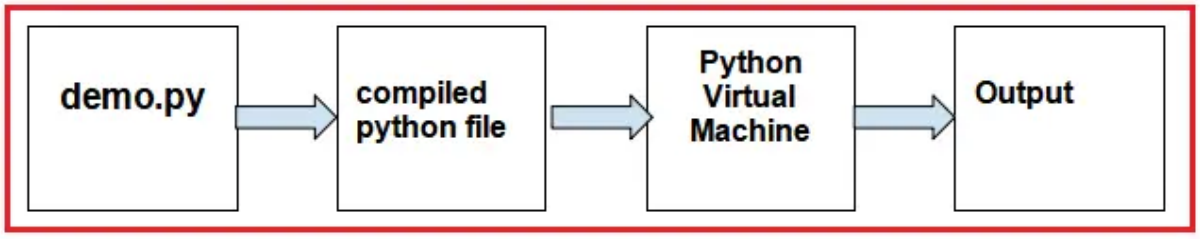
Output: HELLO WORLD

Understanding demo1.py

print("HELLO WORLD"): print() is a predefined function/method in python. It takes strings values as input parameters. It prints the output on the console.

Python Program Execution Flow

The below flow is followed during the execution



Python Program Execution Flow - Python Coding Instructions

First, we need to write a code and save the file with a .py extension. Then we need to run the program. When the program is running, the Internal python compiler takes the source code and creates the corresponding compiled python file, which is not visible to the user The compiled file is saved in the computer cache. If you want to see the file, then you can run the below command

python -m py_compile demo.py

Here, the -m argument states that it's a module, and the module name is py_complile. This is the module that creates the compiled file which is stored as __pycache__ in the same directory.

Now the compiled file contains byte code instructions that are not understandable by the microprocessor to generate output. So, the Python Virtual Machine comes into the picture and takes the responsibility of converting the byte code instructions to machine-understandable format and it does this line by line. Finally, we will see the output

PVM uses an interpreter which converts line by line, which is very slow. To resolve this, some flavors like PyPy use compilers like JIT(Just in Time) which convert very fast.