



Python-A Research On Its Concepts, Modules and Frameworks

Rajesh Babu Nagineni¹

¹Assistant Professor, Dept. Of Computer Science and Engineering, KLEF, Vaddeswaram, India
rajeshnagineni2010@gmail.com

ABSTRACT

Python is a High Level and Multi Paradigm Programming Language, Which was designed and developed by “Guido Van Rossum” in the late 1980’s. Python is easy to learn language and very apt for Real World Programming. Python is having its importance in the real world domains such as Web Development, Game Development, Machine Learning, Artificial Intelligence, Data Science, Desktop GUI Apps, Web Scraping Applications, Business Applications, Audio and Video Applications, CAD Applications and Embedded Applications. The use of Python in Large no of domains is due to its simplicity of language constructs, Multi Paradigm nature of language, vast no of modules and large no of Frameworks available. In this Paper I am going to discuss about the Python’s Concepts, Modules and Frameworks I am also discussing about the Merits and Limitations of Python.

Key words: Python, Programming Language, Real World Programming Modules, Frameworks

1. INTRODUCTION

Python is a General Purpose, High Level, Dynamic and Multi Paradigm Language, which is apt for Beginners of programming to Real World Programming. The Popularity of Python is due to its Simple Syntax, Dynamic Typing, Automatic memory Management and facilitating of programming in multiple programming paradigms Such as Procedural, Imperative, Object Oriented and Functional. The wide use of Python in vast no of domains such as Gaming, Data Science, Machine Learning, GUI Applications, Web Applications, Images, Robotics and Embedded Applications is due to its large no of modules and high no of open source frameworks. Python is designed and developed in a way to make the programmer to

focus on the logic rather than on the complexity of the programming constructs, this is possible by Python which needs few lines of code to develop a program/project. All this is Possible due to the built in and third party modules of Python. This feature of Python is in contrast of Programming Languages such as C, C++ and Java, which will Use more no of lines of code to develop a program/project. Python is a Interpreted Language unlike C and C++. The Programmers need not compile the Python Programs, instead they straight away interpret the programs. Python internally Compiles and interprets the program and gives out the results. This is the one of the Powerful features of the Python. More over Python interpreters are available for all Major Operating Systems and Platforms.

2. CHARACTERISTICS OF PYTHON

Python comes with a wide variety of Characteristics. They are

(a) Syntax

Python comes with an Elegant syntax that facilitates the programmers to write and read the programs easily

(b) Easy to Use

Python is designed in such a way that it can be used for prototype development and real world programming without compromising on maintainability of code

(c) Large Standard Library

Python comes with large no of libraries and modules that can be used for common Programming tasks such as connecting to servers, searching text with regular Expressions, reading and modifying files

(d) Built in IDE

Python is bundled with built in IDE called as IDLE (Integrated Development and Learning

Environment). This IDLE can be used as interactive mode for the programmers to test and debug small snippets of code.

(e) Cross Platform Support

Python "Runs Anywhere", means Python is having interpreters for all of the major Platforms. Even it is having interpreters for unofficial builds of Android and IOS

(f) Support for variety of Data Types

Python supports a large no of Data types Such as Numbers, unlimited length Long Integers floats, complex numbers, String for both ASCII And Unicode, lists, tuples, Sets and Dictionaries

(g) Code Modules

Python code can be grouped into modules and packages

(h) Built in Exception Handling

Python is having built for support for handling of exceptions. This result in cleaner error handling

(I) Dynamic Typing

Python data types are strongly and dynamically typed so merging of incompatible Types causes an exception which can be caught easily

(j) Advanced Programming Features

Python comprises of advanced programming features such as Generators and List Comprehensions

(k) Built in support for Memory Management

Python's Automatic Memory Management frees the programmers from being allocate and free up the memory for the programming constructs

3.FEATURES OF PYTHON

Python is a High Level, Multi Paradigm Programming Language, designed to suits the needs of the Beginners of the programming and to develop the large scale real world projects and products. Python is having the features that

matches the requirements of the various real time domains such as machine learning, deep learning, data science, robotics, game development

embedded systems. Python is having the below stated features.

(a) Easy Language

Python is an easy language. It is easy to read, write, learn and understand. Python has a smooth learning curve. It is easy to learn. Python has a simple syntax and Python code is easy to understand. Since its easy to understand, you can easily read and understand someone else's code. Python is also easy to write because of its simple syntax. Because it is an easy language, it is used in schools and universities to introduce students to programming. Python is for both start ups and big companies.

(b) Readable

The Python language is designed to make developers life easy. Reading a Python code is like reading an English sentence. This is one of the key reason that makes Python best for beginners. Python uses indentation instead of curly braces, unlike other programming languages. This makes the code look clean and easier to understand.

(c) Interpreted Language

Python is an interpreted language. It comes with the IDLE (Interactive Development Environment). This is an interpreter and follows the REPL structure (Read-Evaluate-Print-Loop). It executes and displays the output of one line at a time. So it displays errors while you're running a line and displays the entire stack trace for the error.

(d) Dynamically Typed Language

Python is not statically-typed like Java. You don't need to declare data type while defining a variable. The interpreter determines this at runtime based on the types of the parts of the expression. This is easy for programmers but can create runtime errors. Python follows duck-typing. It means, "If it looks like a duck, swims like a duck and quacks like a duck, it must be a duck."

(e) Multi Paradigm Language

Python supports Multiple Programming Paradigms to develop the programs/projects. Some of the paradigms are Procedural, Object Oriented, Imperative and Functional. Python supports the Object Oriented features like Inheritance, Encapsulation and Polymorphism.

(f) Popular and Large Community Support

Python Language is popular and having large community support on the Internet. Python is having its significance in the communities such as Stack Overflow and Meetup. These communities will answer your queries. Already there are many answered queries in these communities.

(g) Open Source

Python is Free and Open Source, means Python is freely available over the Internet. For Download, Python allows the programming community to make changes to the language and again againt distributed over Internet, This feature makes the Python to get better day by day.

(h) Large Standard Library

The standard library is large and has many packages and modules with common and important functionality. If you need something that is available in this standard library, you don't need to write it from scratch. Because of this, you can focus on more important things. You can also install packages from the PyPI (Python Package Index) if you want even more functionality.

(i) Platform Independent

Python is a platform independent programming language. That means Python runs on all major platforms such as Windows, Linux, MacOS and Solaris. There is no need to write program separately for each platform.

(j) Extensible and Embeddable

Python is both Extensible and Embeddable. Extensibility means it is possible to extend the code of native languages such as C and C++ to Python. That means you can add the code of C and C++ in Python code. Embeddability means adding the code of Python in the native code such as C and C++.

(k) GUI Support

Python is also used to develop Graphical User Interface (GUI) Applications. Python is having built in modules such as TkInter, PyQt, wxPython and pySide to implement the GUI Applications.

(l) High Level Language

Python is a high-level language and C is mid-level. Python is easy to understand and closer to the user. You don't need to remember system architecture or manage the memory.

4. APPLICATIONS OF PYTHON

Due to the Elegant Syntax and Simple programming constructs, Python is used in a wide variety of Domains. Some are described below.

(a) Web Development

Python offers numerous options for web development. For instance, you have Django, Pyramid, Flask, and Bottle for developing web frameworks and even advanced content management systems like Plone and Django CMS. These web frameworks are packed with standard libraries and modules which simplify tasks like content management, database interaction, and interfacing with internet protocols like HTTP, SMTP, XML, JSON, FTP, IMAP, and POP. Python Web Frameworks are known for their security, scalability, and flexibility. To add to that, Python's Package Index comes with useful libraries like Requests, BeautifulSoup, Paramiko, Feedparser, and Twisted Python.

(b) Game Development

Python comes loaded with many useful extensions (libraries) that come in handy for the development of interactive games. For instance, libraries like PySoy (a 3D game engine that supports Python 3) and PyGame are two Python-based libraries used widely for game development. Python is the foundation for popular games like Battlefield 2, Frets on Fire, World of Tanks, Disney's Toontown Online, Vega Strike, and Civilization-IV. Apart from game development, game designers can also use Python for developing tools to simplify specific actions such as level design or dialog tree creation, and even use those tools to export those tasks in formats that can be used by the primary game engine. Also, Python is used as a scripting language by many game engines.

(c) Scientific and Numeric Applications

Python has become a crucial tool in scientific and numeric computing. In fact, Python provides the skeleton for applications that deal with computation and scientific data processing. Apps like FreeCAD (3D modeling software) and Abaqus (finite element method software) are coded in Python.

(d) Artificial Intelligence and Machine Learning

AI and ML models and projects are inherently different from traditional software models. When we talk about AI/ML projects, the tools and technologies used and the skill set required is totally different from those used in the development of conventional software projects. AI/ML applications require a language that is stable, secure, flexible, and is equipped with tools that can handle the various unique requirements of such projects. Python has all these qualities, and hence, it has become one of the most favored languages of Data Science professionals. Python's simplicity, consistency, platform independence, great collection of resourceful libraries, and an active community make it the perfect tool for developing AI and ML applications.

(e) Desktop GUI

Python not only boasts of an English-like syntax, but it also features a modular architecture and the ability to work on multiple operating systems. These aspects, combined with its rich text processing tools, make Python an excellent choice for developing desktop-based GUI applications. Python offers many GUI toolkits and frameworks that make desktop application development a breeze. PyQt, PyGtk, Kivy, Tkinter, WxPython, PyGUI, and PySide are some of the best Python-based GUI frameworks that allow developers to create highly functional Graphical User Interfaces (GUIs).

(f) Software Development

Python packages and applications aim to simplify the process of software development. From developing complex applications that involve scientific and numeric computing to developing desktop and web applications, Python can do it all. This is the reason why Software Developers use Python as a support language for build control, testing, and management.

(g) Enterprise Level/Business Applications

Python packages and applications aim to simplify the process of software development. From developing complex applications that involve scientific and numeric computing to developing desktop and web applications, Python can do it all. This is the reason why Software Developers use Python as a support language for build control, testing, and management. Thus, these applications must be capable of integrating with legacy systems like existing databases and non-web apps. Since business applications are developed, keeping in mind the custom requirements to cater to the specific needs of an organization's operating model, the entire development process becomes very complicated. This is where Python can make a significant difference. Python high performance, scalability, flexibility, and readability are just the features required for developing fully-functional and efficient business applications.

(h) Education Programs and Training Courses

This is where Python can make a significant difference. Python high performance, scalability, flexibility, and readability are just the features required for developing fully-functional and efficient business applications. However, Python is not just great as an introductory language even professional developers and coders all around the world rely heavily on Python.

(i) Language Development

Over the years, Python's design and module architecture has been the inspiration behind the development of many new programming languages such as Boo, Swift, CoffeeScript, Cobra, and OCaml. All of these languages share numerous similarities with Python on grounds like object model, syntax, and indentation.

(j) Operating Systems

Python is the secret ingredient behind many operating systems as well, most popularly of Linux distributions. Linux-based Ubuntu's Ubiquity Installer and Fedora and Red Hat Enterprise's Anaconda Installer are coded in Python. Even Gentoo Linux leverages Python Portage (package management system). Usually, Python is combined with the C programming language to design and develop operating systems.

(k) Web Scrapping Applications

Python is a nifty tool for extracting voluminous amounts of data from websites and web pages. The pulled data is generally used in different real-world processes, including job listings, price comparison, R&D, etc. BeautifulSoup, MechanicalSoup, Scrapy, LXML, Python Requests, Selenium, and Urllib are some of the best Python-based web scraping tools.

(l) Image Processing and Graphic Design Applications

Alongside all the uses mentioned above, Python also finds a unique use case in image processing and graphic design applications. The programming language is used globally to design and build 2D imaging software like Inkscape, GIMP, Paint Shop Pro, and Scribus. Also, Python is used in several 3D animation packages such as Blender, Houdini, 3ds Max, Maya, Cinema 4D, and Lightwave, to name a few.

5. MODULES of Python

Python's main strength relies on its Simple Syntax, Simple Programming constructs and its built-in Modules. Python is having large no of modules which facilitates the Programming community to develop code for different variety of domains such as Data Science, Machine Learning, Deep Learning, Robotics, Gaming, Web Development, GUI Applications, System Programming, Enterprise Applications and Embedded Systems. The Role of Python in the above mentioned domains is Due to its simplified syntax, clear programming constructs and object oriented nature of the language. Python's modules are classified into two types, they are Built-in Modules and Third Party Modules.

Built in Modules:**OS Module**

OS module facilitates the users to perform many operating system tasks. This module is having functions for creating and removing a directory, fetching the contents and changing and identifying the current directory. Some of the functions of this module are: `mkdir()` to create a new directory, `chdir()` to change the current working directory, `rmdir()` to removing a directory, `listdir()` to list all the directories and files in the specified directory

Sys Module

Sys Module allows the programmers to manipulate the different parts of the Python runtime environment. This is one of the important modules of Python. This module contains functions such as `sys.argv`, returns a list of command line arguments passed to a Python program, `sys.exit` causes the exit back to either the Python console or the command prompt. `sys.maxsize` returns the largest integer a variable can take. `sys.path` is an environment variable that is search path for all Python modules. `sys.version` displays the version number of the current Python Interpreter.

Math Module

Math module contains some of the most popular mathematical functions. They are Trigonometrical functions, representation functions, logarithmic functions, angle conversion functions are functions. `math.log()` function returns the natural logarithm of a given number. `math.exp()` function returns a float number after raising `e` (`math.e`) to given number. `math.pow()` function receives two floating point arguments raises the first to the second and returns the result. `math.sqrt()` function returns the square root of a given number.

Statistics Module

This module provides functions to mathematical statistics of numerical data. Statistics Module is very useful for Machine Learning Domain. This module provides Functions such as `mean` which is used to find out the arithmetic mean of the numbers in a list. `median` function returns the middle value of numeric data in a list. `Mode` function returns the most common data point in the list. `stdev` function calculates the standard deviation on a given sample in the form of a list.

Collections Module

Collections Module provides alternatives to built-in container data types such as list, tuple and dictionary. This module provides functions such as `namedtuple` function returns a tuple-like object with named fields. `OrderedDict` function remembers the order of the keys in which they were first inserted. `deque` function supports `append` or `pop` the elements from either ends of a list.

Random Module

Random Module contains a pseudo random number generator which generates pseudo random numbers. The functions of this module are random function which generates random floating point numbers between 0.0 and 1.0. randint function returns a random integer between the specified integers. randrange function returns a randomly selected element from the range created by the start, stop and step arguments. choice function returns a randomly selected element from a non empty sequence. shuffle function randomly reorders the elements in a list.

Operator Module

Operator Module provides a set of predefined functions corresponding to operators of Python. This Module contains basic functions such as add(), which adds the two numbers. eq() function compares the two given numbers. abs() function returns the absolute value of the given number. concat() function concatenates or adds the given two numbers. true_div() function performs the integer division of the given two numbers. Pow() function specifies the power of a given number. mod() returns the modulus of the division of the given two numbers.

Decimal Module

Decimal Module provides support for fast correctly rounded decimal floating point arithmetic. This Module contains functions such as adjusted() function which returns the adjusted exponent after shifting the coefficients rightmost digits until only the lead digit remains. as_tuple returns a named tuple representation of the given number. compare function compares the values of two decimal instances. copy_abs function returns the absolute value of the argument.

String Module

String Module provides the common string operations. This module provides some useful string manipulation functions they are, format() function takes a format string and an arbitrary set of positional and keyword arguments. get_value() function retrieves a given field value. substitute() function performs the template substitution and returning a new string. template() function accepts the string as the argument for a constructor.

TKinter Module

Tkinter is a *Python binding* to the *Tk GUI* toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's *de facto standard* GUI. Tkinter is included with standard *Linux*, *Microsoft Windows* and *Mac OS X* installs of Python. The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. Tkinter is *free software* released under a *Python license*.

Third Party Modules

Tensorflow Module

TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models. TensorFlow APIs are arranged hierarchically, with the high-level APIs built on the low-level APIs. Machine learning researchers use the low-level APIs to create and explore new machine learning algorithms. In this class, you will use a high-level API named tf.keras to define and train machine learning models and to make predictions. tf.keras is the TensorFlow variant of the open-source keras API.

PyTorch Module

PyTorch is an open source machine learning library based on the Torch library used for applications such as *computer vision* and *natural language processing*, primarily developed by Facebook's AI Research lab (FAIR). It is *free and open-source software* released under the *Modified BSD license*. Although the *Python* interface is more polished and the primary focus of development, PyTorch also has a *C++* interface. A number of pieces of *Deep Learning* software are built on top of PyTorch, including *Tesla Autopilot*, *Uber's Pyro*, *HuggingFace's Transformers*, *PyTorch Lightning*, and *Catalyst*.

OpenCV Module

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate

the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Theano Module

Theano is a Python Module that allows us to evaluate mathematical operations including multi-dimensional arrays so efficiently. It is mostly used in building Deep Learning Projects. It works a way more faster on Graphics Processing Unit (GPU) rather than on CPU. Theano attains high speeds that gives a tough competition to C implementations for problems involving large amounts of data. It can take advantage of GPUs which makes it perform better than C on a CPU by considerable orders of magnitude under some certain circumstances. This Module knows how to take structures and convert them into very efficient code that uses numpy and some native libraries. It is mainly designed to handle the types of computation required for large neural network algorithms used in Deep Learning. That is why, it is a very popular library in the field of Deep Learning.

Selenium Module

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain specific language (Selenese) to write tests in a number of popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium runs on Windows, Linux,

and macOS. It is open-source software released under the Apache License 2.0.

Web2py Module

Web2py is an *open-source web application framework* written in the *Python programming language*. Web2py allows *web developers* to program *dynamic web content* using *Python*. Web2py is designed to help reduce tedious web development tasks, such as developing *web forms* from scratch although a web developer may build a *form* from scratch if required. Web2py was originally designed as a teaching tool with emphasis on ease of use and *deployment*. Therefore, it does not have any project-level configuration files. The design of web2py was inspired by the *Ruby on Rails* and *Django* frameworks. Like these frameworks, web2py focuses on *rapid development*, favors *over configuration* approach and follows a *model-view-controller (MVC) architectural pattern*.

Pylons Project Module

Pylons Project is an *open-source* organization that develops a set of web application technologies written in *Python*. Initially the project was a single web framework called Pylons, but after the merger with the repoze.bfg framework under the new name Pyramid, the Pylons Project now consists of multiple related web application technologies.

Chainer Module

Chainer is an *open source deep learning* framework written purely in *Python* on top of *NumPy* and *CuPy* Python libraries. The development is led by Japanese venture company Preferred Networks in partnership with *IBM*, *Intel*, *Microsoft*, and *Nvidia* and *Nvidia*. Chainer is notable for its early adoption of "*define-by-run*" scheme, as well as its performance on large scale systems. The first version was released in June 2015 and has gained large popularity in Japan since then. Furthermore, in 2017, it was listed by *KDnuggets* in top 10 open source machine learning Python projects.

Numpy Module

NumPy is a library for the *Python programming language*, adding support for large, multi-dimensional *arrays* and *matrices*, along with a

large collection of high level mathematical to operate on these arrays.[5] The ancestor of NumPy, Numeric, was originally created by *Jim Hugunin* with contributions from several other developers. In 2005, *Travis Oliphant* created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is *open-source software* and has many contributors.

SymPy Module

SymPy is an *opensource Python library* for *symbolic computation*. It provides computer algebra capabilities either as a standalone application, as a library to other applications, or live on the web as *SymPy Live* or *SymPy Gamma*. SymPy is simple to install and to inspect because it is written entirely in Python with few dependencies. This ease of access combined with a simple and extensible codebase in a well known language make SymPy a computer algebra system with a relatively low barrier to entry. SymPy includes features ranging from basic symbolic arithmetic to calculus, algebra, *discrete mathematics* and *quantum physics*. It is capable of formatting the result of the computations as *LaTeX* code.

SQLAlchemy Module

SQLAlchemy is an *open-source SQL* toolkit and *object-relational mapper (ORM)* for the *Python programming language* consists language. SQLAlchemy's philosophy is that relational databases behave less like object collections as the scale gets larger and performance starts being a concern, while object collections behave less like tables and rows as more abstraction is designed into them. For this reason it has adopted the *data mapper pattern* (similar to *Hibernate* for *Java*) rather than the *active record pattern* used by a number of other object-relational mappers. However, optional plugins allow users to develop using declarative syntax.

Python Imaging Library

Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a *free and open-source additional library* for the *Python programming language* that adds support for opening, *manipulating*, and saving many different *image file formats*. It is available for *Windows*, *Mac OS X* and *Linux*. The latest

version of PIL is 1.1.7, was released in September 2009 and supports Python 1.5.2–2.7, with *Python 3* support to be released "later". Development appears to be discontinued, with the last commit to the PIL *repository* coming in 2011. Consequently, a successor project called Pillow has *forked* the PIL repository and added Python 3.x support.

6.FRAMEWORKS OF PYTHON

Python Frameworks or Integrated Development Environments (IDEs) are the Development environments for Python and some other popular programming languages. There are different variety of Frameworks are available for Python They are described below

IDLE of Python

Integrated Development and Learning Environment (IDLE) is the basic IDE for Python It is integrated in to the Python Language. Python IDLE contains an Interpreter, which is used to run and test small programs. This is the first IDE of Python programming language.

PyCharm

PyCharm is a cross-platform IDE used for Python programming. This editor can be used on Windows, macOS, and Linux. This software contains API that can be used by the developers to write their own Python plugins so that they can extend the basic functionalities.

Spyder

Spyder is a scientific integrated development environment written in Python. This software is designed for and by scientists who can integrate with Matplotlib, SciPy, NumPy, Pandas, Cython, IPython, SymPy, and other open-source software. Spyder is available through Anaconda (open-source distribution system) distribution on Windows, macOS, and Linux.

Atom

Atom is a useful code editor tool preferred by programmers due to its simple interface compared to the other editors. Atom users can submit packages and them for the software.

Jupyter

Jupyter is a tool for people who have just started with data science. It is easy to use, interactive data science IDE across many programming

languages that just not work as an editor, but also as an educational tool or presentation.

PyDev

PyDev is a third-party Python editor for Eclipse. This editor can be used in not only Python but IronPython and Jython development.

Thonny

Thonny is an IDE for learning and teaching programming, specially designed with the beginner Pythonista scripting environment. It is developed at The University of Tartu, which you can download for free on the Bitbucket repository for Windows, Linux, and Mac

Wing

Wing is a lightweight Python environment which is designed to give you productive development experience. Wing supports test-driven development with various frameworks like the unittest, pytest, nose, doctest, and Django testing.

ActivePython

Increase software development data science with a secure and supported Python distribution. ActivePython is software consisting of the Python implementation CPython and a set of various extensions to facilitate installation.

Anaconda

Anaconda is a *free and open-source* distribution of the *Python* and *R* programming languages for *scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)*, that aims to *simplify package management* and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and *Travis Oliphant* in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, which are both not free.

7.MERITS OF PYTHON

(a) Presence of Third Party Modules

The Python Package Index (PyPI) contains numerous third-party modules that make Python capable of interacting with most of the other languages and platforms.

(b) Extensive Support Libraries

Python provides a large standard library which includes areas like internet protocols, string operations, web services tools and operating system interfaces. Many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly.

(c) Open Source and Community Development

Python language is developed under an OSI-approved open source license, which makes it free to use and distribute, including for commercial purposes. Further, its development is driven by the community which collaborates for its code through hosting conferences and mailing lists, and provides for its numerous modules.

(d) Learning Ease and Support Available

Python offers excellent readability and uncluttered simple-to-learn syntax which helps beginners to utilize this programming language. The code style guidelines, PEP 8, provide a set of rules to facilitate the formatting of code. Additionally, the wide base of users and active developers has resulted in a rich internet resource bank to encourage development and the continued adoption of the language.

(e) User Friendly Data Structures

Python has built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, Python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

(f) Productivity and Speed

Python has clean object-oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity. Python is considered a viable option for building complex multi-protocol network applications

8. LIMITATIONS OF PYTHON

(a) Slow Execution Speed

Python is an interpreted language, which means it works with an interpreter, not with a compiler. As a result, it executes relatively slower than C, C++, Java, and many other languages.

(b) Large Memory Consumption

Python's structures demand more memory space. This language is not suitable to use for development under limited memory restrictions.

(d) Not Suitable for Mobile and Game Development

Python is mostly used in desktop and web server-side development. It is not considered ideal for mobile app development and game development due to the consumption of more memory and its slow processing speed while compared to other programming languages.

(d) Developers Restrictions

Once a developer gets used to the ease and simplicity of this language, it becomes difficult for them to switch back to other languages.

(e) Error Detection in Codes

As Python is executed through an interpreter instead of a compiler, errors and bugs can't be detected during compilation, and that's not good for developers.

(f) Database Access

Python is considered to be highly insecure and involves security risk. There are some limitations to using Python to access databases. As compared to other popular technologies such as JDBC and ODBC, the Python database access layer is a little underdeveloped and primitive. It is, therefore, not considered suitable if developers are looking for a smooth interaction of complex legacy data.

9. CONCLUSION

By this Research on Python It is concluded that Python is one of the vastly used Programming Languages because of its Elegant Syntax, Simple Programming constructs, rich set of Modules and Frameworks. Despite of all Merits and Limitations of Python, It is still one of the most preferred and trusted languages for Desktop and Web Development. Python is

best suited for Machine Learning and Artificial Intelligence and it is not mostly suitable for Mobile Development and Game Development. I finally concluded that the merits of Python overcomes its Limitations. If Python is used in the right way by a skilled developer, Python is among the best choices for coding.

REFERENCES

- [1] Summerfield, Mark. Rapid GUI Programming with Python and Qt.
- [2] "Programming Language Trends - O'Reilly Radar". Radar.oreilly.com. 2 August 2006.
- [3] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises".
- [4] Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming
- [5] <https://python.org>
- [6] Automate the Boring Stuff with Python: Practical Programming for Total Beginners
- [7] <https://techvidvan.com>
- [8] <https://upgrad.com>
- [9] <https://tutorialsteacher.com>
- [10] <https://faceprep.in>
- [11] <https://hackr.io>
- [12] <https://invensis.net>